NPS-CS-07-007

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

System Requirements Analysis and Technological Support for the Ballistic Missile Defense System (BMDS)

FY07 Progress Report

By
M. Auguston, D. Drusinsky, R. Hutchins J.B. Knorr,
J.B. Michael, T. Otani, P.E. Pace, M. Shing, M. Tummala,
T. Cook, P. Katopodis, T.O. Walker, Y.Q. Chen, G. Katsis,
D. Little, D. Patsikas, Z.P. Pace, B. Rakdham,
A.J. Sampson, H. Tummala, C.C. Waddell

July 2007

Approved for public release; distribution is unlimited

Prepared for: Missile Defense Agency
7100 Defense Pentagon
Washington, D.C. 20301-7100

THIS PAGE INTENTIONALLY LEFT BLANK

NAVAL POSTGRADUATE SCHOOL
Monterey, California 93943-5000

Daniel T. Oliver
President

Leonard A. Ferrari
Provost

This report was prepared for the Missile Defense Agency and funded by the Missile Defense Agency.

This report was prepared by:

____
James Bret Michael
Professor of Computer Science and Electrical Engineering
Naval Postgraduate School

Reviewed by:

____
Peter J. Denning, Chairman
Department of Computer Science

Released by:

____
Dan C. Boger
Interim Associate Provost and
Dean of Research

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE *Form Approved OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** July 2007	**3. REPORT TYPE AND DATES COVERED** Technical Report

4. TITLE AND SUBTITLE: Title (Mix case letters)
System Requirements Analysis and Technological Support for the Ballistic Missile Defense System (BMDS) - FY07 Progress Report

5. FUNDING NUMBERS
MD7080101P0630

6. AUTHOR(S) M. Auguston, D. Drusinsky, R. Hutchins J.B. Michael, J.B. Knorr, T. Otani, P.E. Pace, M. Shing, M. Tummala, T. Cook, P. Katopodis, T.O. Walker, Y.Q. Chen, G. Katsis, D. Little, D. Patsikas, Z.P. Pace, B. Rakdham, A.J. Sampson, H. Tummala, C.C. Waddell

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER NPS-CS-07-007

9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Missile Defense Agency, 7100 Defense Pentagon, Washington, DC 20301-7100

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this report are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (maximum 200 words)

Engineering of the Ballistic Missile Defense System (BMDS) requires one to take a holistic approach that includes the physical modeling and analysis of the missile defense operating environment, development of metrics and techniques to analyze the communication requirements of the net-centric Ballistic Missile Defense warfare, and the use of architectural patterns and other software technologies to shape the emergent behavior of the BMDS taking into account of the system's interoperability, composability, extensibility, and dynamic reconfigurability. This report summarizes the work in FY07 to investigate new technologies to support the development of the BMDS. We developed new scoring functions for the fusion of sensor data, an algorithm for multiple hypothesis tracking, a distributed medium access control protocol and data dissemination algorithm for wireless networks of cooperative radar systems, simulation models for network-centric electronic warfare metrics study and for the prediction of the over the horizon radar system footprints, technologies for the correct specification and validation of temporal behaviors in a Service-Oriented Architecture (SOA) based system-of-systems, runtime verification of system-level requirements of distributed reactive systems using MSC-Assertions, and safety assurance of reconfigurable and self-reconfigurable systems. We also evaluated the effectiveness of the real-time Java technology for BMDS software and the potential impact of integrating the Air Force YAL-1A Attack Laser into the BMDS.

14. SUBJECT TERMS
Real-time system, Java programming language, Real-Time garbage collection, Ballistic Missile Defense System, Global Integrated Fire Control, Advanced Battle Manager, Tracking, Sensor networking, System-of-Systems

15. NUMBER OF PAGES
61

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

1 Introduction 1

2 Safety Assurances of Reconfigurable Systems 2

2.1 Safety Requirements of the GIFC 2

2.2 Safety Assurance of the GIFC 3

2.2.1 Safety Assertions 3

2.2.2 Run-time Monitoring of Safety Assertions 4

2.2.3 Safety Wrappers 4

2.2.4 Safety Kernels 5

3 Service-Oriented Architecture Based System-of-Systems 6

3.1 Message Sequence Chart Assertions 6

3.2 Validation and Execution Monitoring of MSC Assertions 7

4 Verifying Distributed Protocols 8

4.1 OMNeT++ and UML Models 8

4.2 System-Level Simulation and Scenario Generation 10

5 Experiments with Real-Time Java 11

5.1 RTS v2.0 Real-Time Garbage Collector 12

5.2 Experiments with RTS2.0 12

5.2.1 Experiments 1 to 4 12

5.2.2 Experiment 5 - Interaction between RTGC and Realtime Threads 13

5.2.3 Experiment 6 - Mixture of Critical and Non-Critical Real-Time Threads 15

5.2.4 Experiment 7 - Modified Experiment 4 16

6 Data Fusion and Tracking 18

7 Ballistic Missile Simulation and Tracking 19

7.1 Results of Simulation Comparison 21

8 Multiple Hypothesis Tracking 23

8.1 Functional Areas 24

8.1.1 Sensor Modeling 24

8.2 Multiple Hypothesis Tracking – Linear Assignment Approach 25

8.3 Experimental Results 26

9 Distributed Medium Access and Data Dissemination for Cooperative Radar Systems 29

9.1 CR-MAC: A Distributed Medium Access Control Protocol for Wireless Networks of Cooperative Radar Systems 29

9.2 Data Dissemination Algorithm for a Hybrid Large-Scale Wireless Sensor Network 31

9.3 Multistage Network Security 35

9.4 Sensor Motes for Target Tracking 35

10 Missile Defense Technologies 35
10.1 Predicting Threat OTH Radar Footprints 35
10.2 Electronic Warfare Metrics 36
11 Attack Laser ABM 37
12 Conclusion 39
12.1 High-Assurance Requirements Specification, Validation and Verification Techniques 39
12.2 Run-time Network Security Monitoring 40
12.3 Real-Time Java System Evaluation 40
12.4 Data Fusion 40
12.5 Ballistic Missile Tracking 41
12.6 Sensor Networks 42
12.7 Simulation of Electronic Warfare Metrics 43
13 References 45
14 List of Acronyms 48
Initial Distribution List 51

1 Introduction

The Ballistic Missile Defense System (BMDS) is a complex distributed system-of-systems (SoS) composed of sensors, weapons, C2BMC software, and other components, some of which are non-developmental or even non-organic to the BMDS. Engineering of the BMDS requires one to take a holistic approach that includes the physical modeling and analysis of the missile defense operating environment, development of metrics and techniques to analyze the communication requirements of the net-centric Ballistic Missile Defense warfare, and the use of architectural patterns and other software technologies to shape the emergent behavior of the BMDS taking into account of the system's interoperability, composability, extensibility, and dynamic reconfigurability..

This report summarizes the work in FY07 to investigate new technologies to support the development of the BMDS. The research was conducted by nine faculty, three doctoral students, five master's students, and four summer interns at the Naval Postgraduate School (NPS). We developed new scoring functions for the fusion of sensor data, an algorithm for multiple hypothesis tracking, a distributed medium access control protocol and data dissemination algorithm for wireless networks of cooperative radar systems, simulation models for network-centric electronic warfare metrics study and for the prediction of the over the horizon radar system footprints, technologies for the correct specification and validation of temporal behaviors in a Service-Oriented Architecture (SOA) based SoS, runtime verification of system-level requirements of distributed reactive systems using MSC-Assertions, and safety assurance of reconfigurable and self-reconfigurable systems. We also evaluated the effectiveness of the real-time Java technology for BMDS software, and the potential impact of integrating the Air Force YAL-1A Attack Laser into the BMDS.

The report is organized as follows. Section 2 discusses the software technologies for the safety assurances of reconfigurable systems. Section 3 presents a new formalism, called the Message Sequence Chart (MSC) assertion, for specifying temporal behavior in systems-of-systems (SoSes) and the validation and enforcement of these assertions via runtime execution monitoring. Section 4 describes the integration of discrete event simulation and runtime execution monitoring of the MSC assertions to verify distributed protocols. Section 5 describes our study on the effectiveness of the real-time Java for the BMDS. Sections 6 presents a scoring function for the fusion of sensor data. Section 7 presents a comparison of hyperbolic and Kalman techniques for tracking a ballistic missile and determining the launch site. Section 8 describes an improved multiple hypothesis tracking algorithm for tracking multiple ballistic missiles, and Section 9 presents a distributed medium access control protocol and data dissemination algorithm for wireless networks of cooperative radar systems. Section 10 describes the simulation models for network centric EW metrics study and for the prediction of the over the horizon radar system footprints, and Section 11 discusses the potential impact of integrating the Air Force YAL-1A Attack Laser into the BMDS. Section 12 finishes the report with concluding remarks and a discussion of future work.

2 Safety Assurances of Reconfigurable Systems [1]

This research is concerned with the safety assurance of reconfigurable and self-reconfigurable systems. A part of the BMDS is the Global Integrated Fire Control System (GIFC) which works as a globally distributed real-time, software-intensive, and safety-critical battle management system. The design of the GIFC provides the capability for the BMDS to dynamically reconfigure itself, in addition to providing for the evolution of the system through the plug-and-play of components and subsystems. The GIFC system must meet the following reconfigurability and self-reconfigurability requirements:

(1) The network of GIFC nodes must be dynamically reconfigurable into any hierarchical structure to support the evolving C2 structures for Ballistic Missile Defense;

(2) The assets (sensors and weapons) assigned to each GIFC node must be dynamically reconfigurable to cope with the changing workload of the BMDS;

(3) The GIFC system must be self-reconfigurable so that when one GIFC node is out of order, its workload and resources (e.g., sensors, weapons, etc.) will be dynamically and automatically re-distributed to other GIFC nodes.

2.1 Safety Requirements of the GIFC

Following the Missile Defense Agency's capability-based acquisition process for development of the GIFC, we started by developing high-level Unified Modeling Language (UML) use cases and then refining these use cases into successively lower level use cases and other UML artifacts (e.g., statecharts). The use cases focus our attention on the "why" and "what" of the system instead of the "how;" that is, use cases are supposed to be implementation-neutral. In conjunction with the Missile Defense National Team (MDNT), we conducted preliminary hazard analyses (PHA) using the UML artifacts as our reference. An example of a hazard we identified from conducting PHA is inadvertent release of weapons. As we identified hazards, we specified safety requirements to address the hazards and recorded those requirements in the safety requirements traceability matrix (SRTM). For those safety requirements that are allocated to software, we record those requirements in the software SRTM (SSRTM). These matrixes are used to trace safety requirements to system hazards and all of the other system artifacts, in support of developing the safety case (i.e., a claim about the level of safety afforded by the system, arguments supporting the claim, and documentation backing up the arguments) for the BMDS that the GIFC poses a tolerable level of safety risk. As the software engineers refine the safety and other system requirements into designs, the safety engineers apply safety analysis techniques such as Failure Modes, Effects and Criticality Analysis (FMECA) to identify potential failure modes, assess the risk associated with those failure modes, to rank the issues in terms of importance and to identify and carry out corrective actions to address the most serious concerns. An example of a causal factor we identified is an error in the weapon-assignment processing that causes one or more erroneous weapons-tasking-order messages to be sent to engagement data stores.

2.2 Safety Assurance of the GIFC

We present four techniques in this section to help assure the safety properties of the GIFC at runtime.

2.2.1 Safety Assertions

Studies have suggested that the process of specifying requirements formally enables developers to gain a deeper understanding of the system being specified, and to uncover requirements flaws, inconsistencies, ambiguities and incompletenesses. Formal specification is key to test automation and run-time error detection and recovery of complex systems. Moreover, the narrower the syntactic and semantic gaps between the models of requirements and design, the more likely the system designers will understand the requirements accurately and build the systems correctly. Figure 1 shows an example of a statechart assertion to assure that GIFC's track processor's workload remains below 75% of it maximum capacity as mush as possible in order to prevent any "erroneous weapon system tasking" fault that may be caused by the unexpected delay in GIFC's sensor track classification.

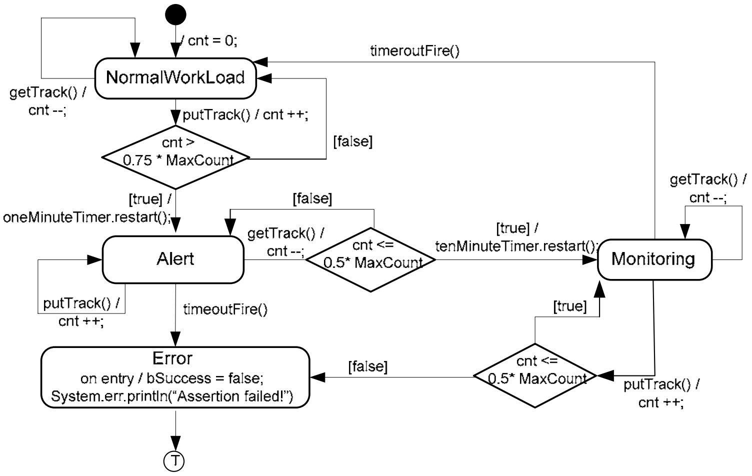


Figure 1: Statechart assertion example

The statechart assertion is written from the point of view of an observer; whenever the statechart assertion observes that the Track Processor receives a *putTrack* event or a *getTrack* event, it updates its variable *cnt* and checks to see if the temporal assertion has been triggered. It uses two timers, *oneMinuteTimer* and *tenMinuteTimer*, to keep track of the temporal conditions of the track processor's workload and will enter the *Error* state if the temporal assertion has been violated.

Most system designers will find the above statechart assertion much more easily understandable than the following equivalent text-based assertion:

```
Always (cnt > 0.75 * MaxCount Implies
   (Eventually <=1 min (Always <=10 min cnt <= 0.5 * MaxCount)))
```

2.2.2 Run-time Monitoring of Safety Assertions

Runtime Execution Monitoring (REM) is a class of methods for tracking the temporal behavior of an underlying application. REM methods range from simple print statement logging methods to run-time tracking of complete formal requirements for verification purposes. Due to the dynamic nature of the BMDS system verification of individual components alone can never give us the kind of assurance one expects to have for the safety-critical BMDS. To increase the robustness of the BMDS we need to armor-plate it against unexpected behaviors. One form of armor-plating is to fortify the software's exception-handling ability via runtime monitoring of temporal assertions, where formal specifications are translated by a code generator into C, C++, or Java statements to be deployed for catching exceptions in the final product during runtime.

2.2.3 Safety Wrappers

The "erroneous weapon system tasking" fault can also be caused by errors due to the dynamic reconfigurable nature of the BMDS. Figure 2 shows a UML-RT model of the high-level architecture of a BMDS, which is made up of a set of GIFC capsules, a set of Weapon_System capsules and a set of Sensor capsules (as indicated by the multi-object icons). In case a fault occurs, the safety wrappers will help detect and isolate the errors traceable to the fault.

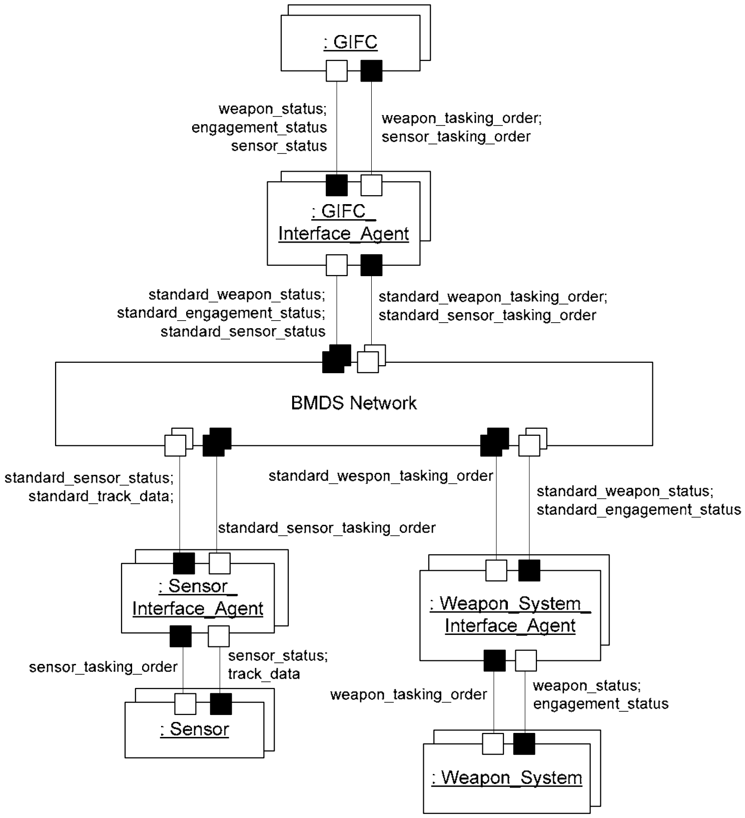


Figure 2: The high-level architecture of a BMDS

2.2.4 Safety Kernels

A safety kernel is a software component of a safety-critical system specifically designed to reduce the probability of occurrence of mishaps. It provides a centralized point for safety processing: the detection, tolerance and isolation of faults that may result in safety hazards. The safety kernel architecture varies from the simple watchdog safety kernel at one end of the spectrum to the dual and multiple redundant safety kernels in the other end of the spectrum. The safety executive is an example of the safety kernel architectures that lie within the middle of the spectrum. The safety executive typically monitors the operation of a system for the following inputs: (1) watchdogs timeouts, (2) exceptions raised by safety-assertion run-time monitors and safety wrappers, and (3) faults from continuous or periodic built-in-tests, and will attempt to recover from the faults by resetting the processes or re-configuring the system via the fail-safe channels, or bring the system to a fail-safe state if the system cannot be fully recovered from the faults.

3 Service-Oriented Architecture Based System-of-Systems [2]

Large SoSes are typically made up of a federation of existing systems and developing systems interacting with each other over a network to provide an enhanced capability greater than that of any of the individual systems within the system-of-systems. Service-oriented architecture (SOA) and the supporting Web Services (WS) technology hold promise to create SoSes that are interoperable, composable, extensible, and dynamically reconfigurable. A Web Service has two components: a contract defining its external behavior from the clients' point of view, and a business process describing its internal logic via the coordination and composition of other Web Services. The specification of these complex business processes is error prone due to concurrency in activities execution, possibility of communication delay and error, as well as faults in the remote service providers. Using light-weight formal methods consisting of the following four steps it is possible to enhance the trustworthiness of the Web Services:

(1) Specify the business process in some semi-formal languages (e.g., the Business Processing Execution Language (BPEL));

(2) Translate the specifications into formal models (e.g., linear temporal logic, state machines, Petri nets, process algebras);

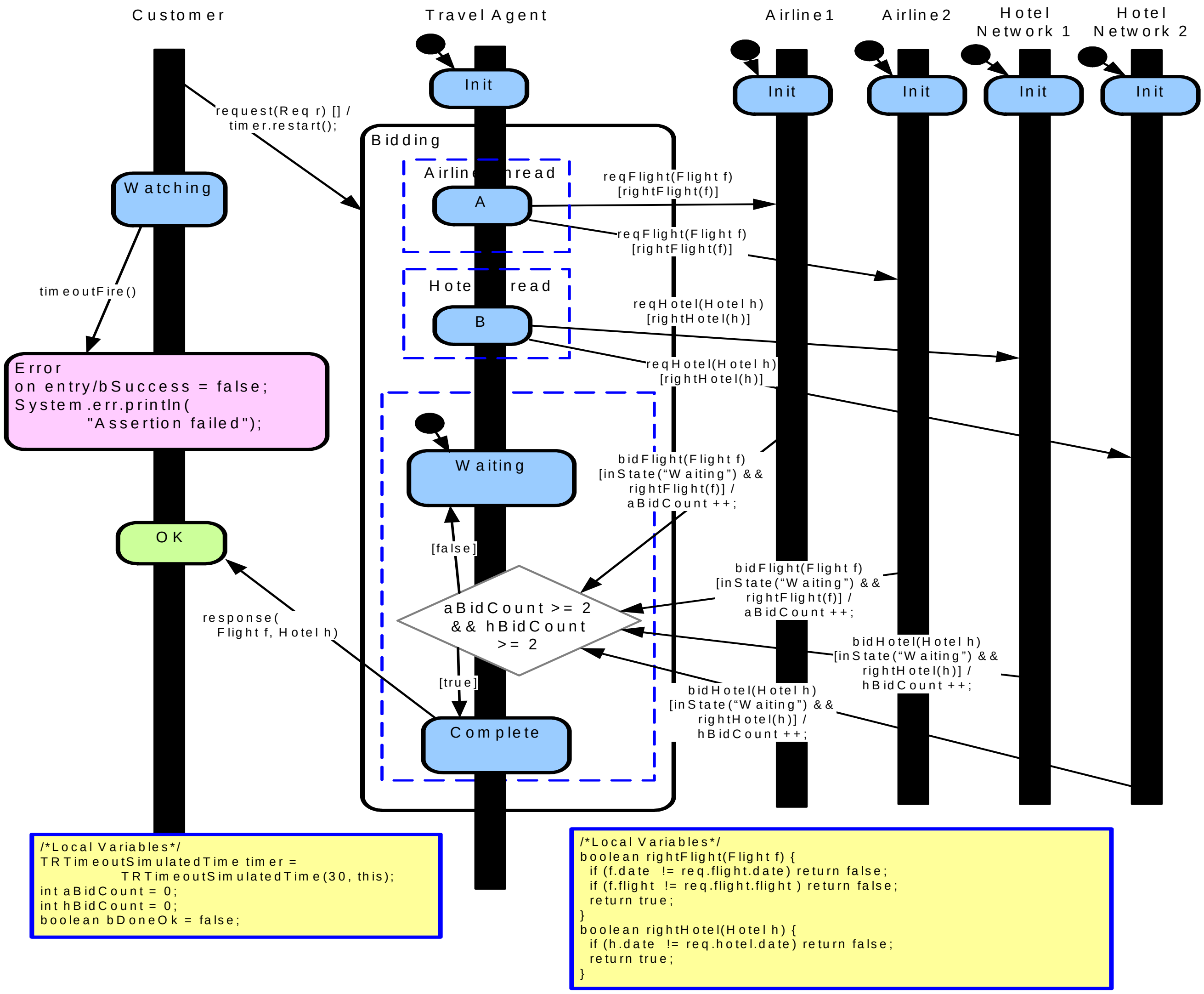
(3) Specify the desirable functional and non-functional properties of the business process as formal assertions;

(4) Verify the formal business process models against the properties by proving theorems, model checking, or specification-based testing.

However, for the aforementioned methods to effectively produce trustworthy SoSes, we need to first validate the accuracy and the correctness of the formal-assertion representation of the mission-essential and safety-critical properties of the SoSes and second verify the business process models against these properties.

3.1 Message Sequence Chart Assertions

MSC Assertions are a formal-language extension of UML Message Sequence Charts (MSC's) superimposed with UML statecharts. MSC Assertions are based on Statechart diagrams superimposed on MSC diagrams and augmented with Java (or C++) conditions and actions. For example, Figure 3 shows the MSC Assertion for a time-bound requirement of a travel agent service: "*R1: The travel agent must obtain bids from at least two airlines and two hotels and return a flight and a hotel matching the customer's request within 30 seconds from the time the customer issues his travel* request."



```
/*Local Variables*/
TRTimeoutSimulatedTime timer =
          TRTimeoutSimulatedTime(30, this);
int aBidCount = 0;
int hBidCount = 0;
boolean bDoneOk = false;
```

```
/*Local Variables*/
boolean rightFlight(Flight f) {
  if (f.date  != req.flight.date) return false;
  if (f.flight  != req.flight.flight ) return false;
  return true;
}
boolean rightHotel(Hotel h) {
  if (h.date  != req.hotel.date) return false;
  return true;
}
```

Figure 3: A MSC assertion for the travel agent service

3.2 Validation and Execution Monitoring of MSC Assertions

It is important to validate the correctness of the assertions early in the software-development process. Unfortunately, users often discover, late in the development process, that their assertions are incorrect and do not work as intended. Our methodology is that requirements be simulated to assure that the cognitive understanding of the requirement matches the formal specification. To that end, we developed a run-time monitor for

MSC Assertions that is fully integrated with the popular JUnit testing framework and created a set of scenarios, using the JUnit testing framework.

Due to the possibility of communication delay and error, as well as faults that may occur during a remote procedure call, verification alone cannot assure the overall level of trustworthiness required of the SOA-based SoSes for BMD applications. To increase the robustness of the system by protecting against unexpected behaviors we can use runtime monitoring of temporal assertions.

4 Verifying Distributed Protocols [3]

The design and implementation of reliable applications on top of asynchronous distributed systems that are prone to processor and network crashes is a difficult and complex task. A distributed system is made up of several components, executing concurrently and interacting with each other under the control of specialized procedures called protocols. Often, distributed-system protocols are correct for an ideal system but do not operate as well in a less than ideal situation. This research addresses the problem using Runtime Execution Monitoring (REM) based techniques. We build upon our previous work on state-chart-assertions and REM of *intra*-agent behavior. Unlike the other approaches, MSC-Assertions, being a natural extension of the statechart assertions, provide a unified model for both intra- and inter-agent behavior specification, thus eliminating the need to translate and maintain two models (one for intra-agent and the other for inter-agent) when designing and analyzing distributed system behaviors. We demonstrate the technique with a proof-of-concept prototype for assessing the failure rate of a time-bound formal requirement for a distributed-system protocol (leader election) operating with non-ideal communication links. The proof-of-concept prototype is as follows:

(1) An OMNeT++ model of a 4-node ring network with parameterized network delays,

(2) A UML statechart model and generated code for the network agents,

(3) A MSC-Assertion for the timely election of a leader and the validation of the assertion via simulation, and

(4) A large set of test scenarios for the 4-node ring distributed system, using white-box test generation techniques.

4.1 OMNeT++ and UML Models

OMNeT++, which stands for *Objective Modular Network Testbed in C++*, is an object-oriented discrete event simulator primarily designed for the simulation of communication protocols, communication networks and traffic models, and multi-processors and distributed systems models. Figure 4 shows a simple OMNeT++ model of a four-node ring network. The network is made up of four identical nodes (agents).

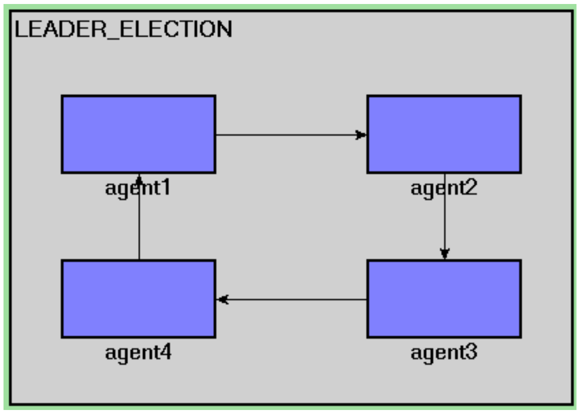
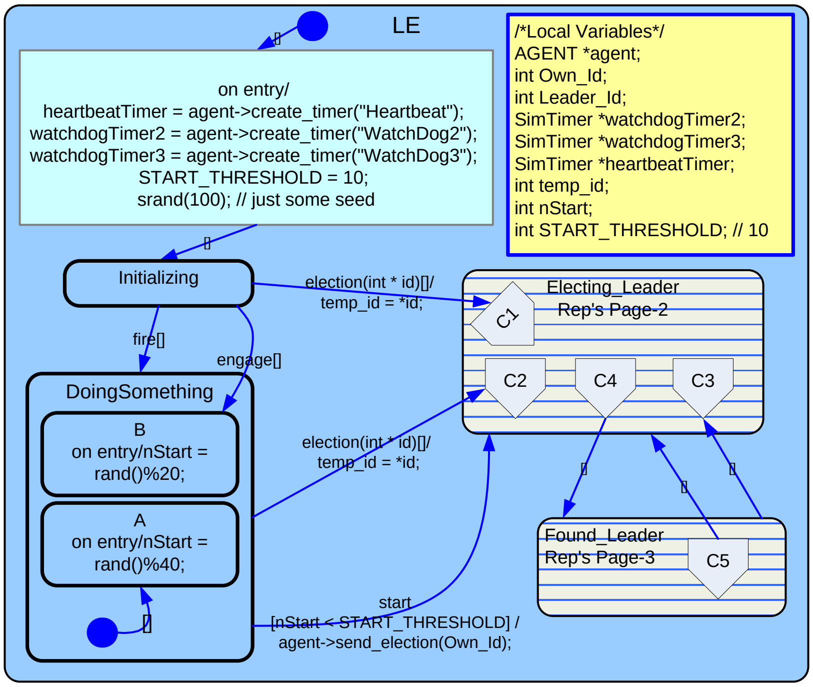


Figure 4: A 4-node ring network model in OMNeT++

Figure 5 shows the top-level statechart of a leader election (*LE*) module. The top-level statechart consists of four states, the *Initializing* state and three composite states named *DoingSomething*, *Electing_Leader* and *Found_Leader*, together with a set of state variables declared in the associated local variable declaration box.



LE
Initializing
fire[]
engage[]
DoingSomething
B
on entry/nStart = rand()%20;
A
on entry/nStart = rand()%40;
election(int * id)[]/ temp_id = *id;
election(int * id)[]/ temp_id = *id;
start [nStart < START_THRESHOLD] / agent->send_election(Own_Id);
Electing_Leader
Rep's Page-2
C1
C2
C4
C3
Found_Leader
Rep's Page-3
C5

```
on entry/
heartbeatTimer = agent->create_timer("Heartbeat");
watchdogTimer2 = agent->create_timer("WatchDog2");
watchdogTimer3 = agent->create_timer("WatchDog3");
START_THRESHOLD = 10;
srand(100); // just some seed
```

```
/*Local Variables*/
AGENT *agent;
int Own_Id;
int Leader_Id;
SimTimer *watchdogTimer2;
SimTimer *watchdogTimer3;
SimTimer *heartbeatTimer;
int temp_id;
int nStart;
int START_THRESHOLD; // 10
```

Figure 5: Top level page of LE statechart

Figure 6 shows a MSC-Assertion for the the following requirement of the 4-node ring: “*All agents contain the same ID for the elected leader, which is the largest identity value among the ID’s of all the active LE modules in the network, in at most 60 seconds after the first election event detected in the network*”.

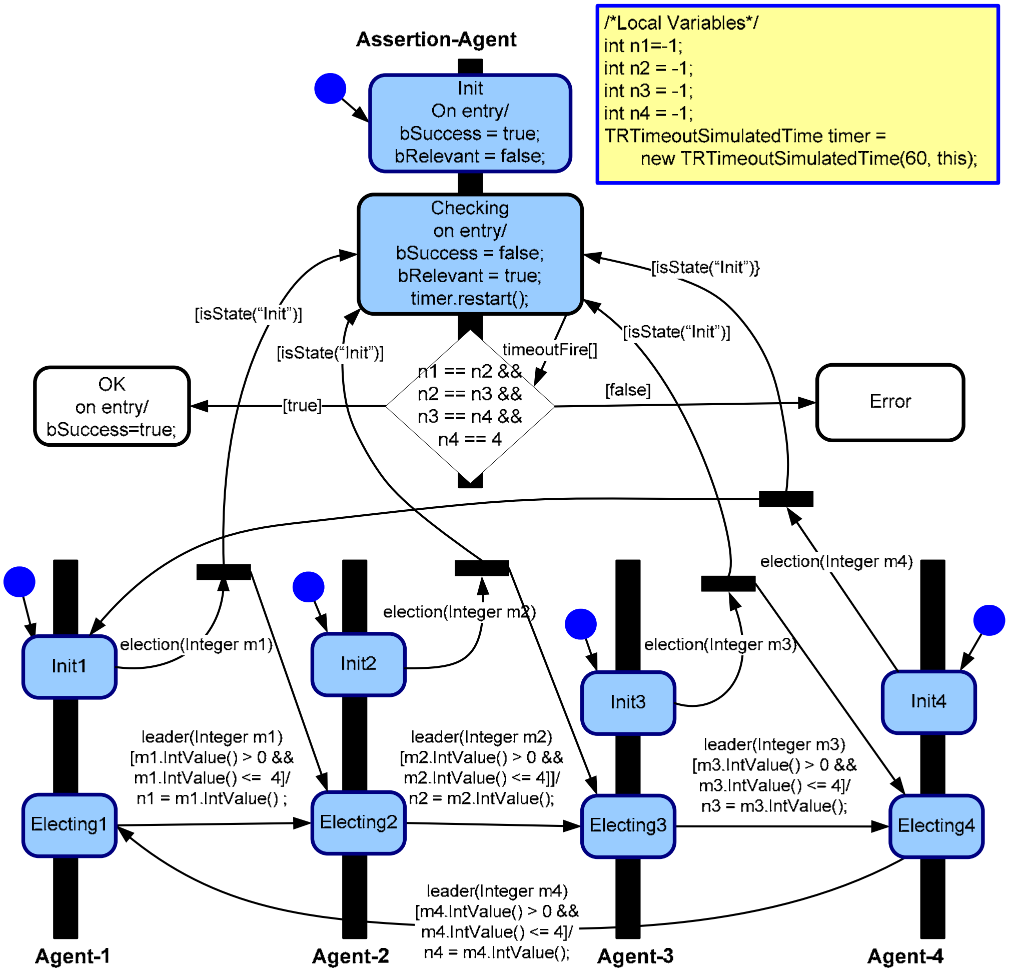


Figure 6: A time-bound leader-election MSC-Assertion for a 4 ring network

4.2 System-Level Simulation and Scenario Generation

In order to assure that there is a cognitive understanding of that the formal specification match the requirements we required a simulation. Automatic scenario generation was constructed using the component level White-Box Automatic Test Generator (WBATG) described in [4] generates a JUnit test suite that repeatedly exercises the component under test using events, time, and data information specified in the component.

The likelihood of success of the leader-election requirement was calculated as the ratio of relevant test scenarios for which the MSC-assertion succeeded to the overall number of relevant tests. With the help of OMNeT++, we can now simulate different network conditions and collect statistics to estimate the likelihood of protocol success.

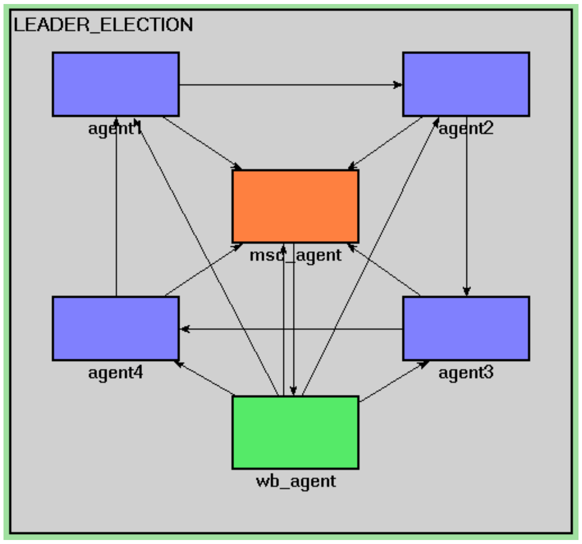


Figure 7: The OMNeT++ model augmented with run-time execution monitor and white-Box Tester

5 Experiments with Real-Time Java[5]

There is an increasing interest in recent years to use the Java^{TM} programming language for implementing real-time systems. Recent advances in the Real-Time Specification for Java (RTSJ) have resulted in the introduction of new means for creating predictable real-time environments for Java programs. However, these new features also make the Java semantics more complex and the run-time behavior of the Java programs more difficult to analyze. In a prior study [6], we concluded that it is preferable to use only the Real-Time Java threads that use the heap memory and not the no-heap real-time threads for the GIFC software, due to the difficulties in writing correct Java programs using the no-heap real-time threads by the majority of the Java programmers in the BMDS project. We proposed a real-time design pattern for a class of real-time applications that allows developers to use (and re-use) Java code libraries and components that use the heap in time-constrained applications. However, the proposed design pattern cannot be implemented by using Sun Java Real-Time System (RTS) 1.0. In FY07, we conducted further experiments to determine if the preferred architecture can be implemented with the Sun Java RTS 2.0, which will give programmers more control over the priority of the garbage collection. The beta version was released in December, 2006 and it supports RTGC. We performed experiments to explore viable software architectures for the GIFC software.

Our experiments indicated that critical improvements made in the beta version did meet our software requirements. In order to run the RTS 2.0 beta version, we are required to update our Solaris machine. The hardware is SunBlade 2500. The operating system is Solaris 10 (11/06) and the RTS 2.0 version is b31 (12/06).

5.1 RTS v2.0 Real-Time Garbage Collector

The real-time garbage collector (RTGC) provided in Sun Java RTS 2.0 has made critical improvements over the previous versions. It is now fully concurrent, and it can be preempted by the application's real-time threads with a higher priority. By properly tuning the values for two runtime parameters—RTGCCriticalPriority and RTGCCriticalReservedBytes—we will be able to achieve the desired balance of deterministic behavior of the critical threads and an overall throughput of the application.

The parameter RTGCCriticalReservedBytes reserves the amount of critical memory. If the amount of free memory becomes lower than RTGCCriticalReservedBytes, the priority of RTGC is increased to RTGCCriticalPriority. This prevents all other threads (non-time-critical real-time threads and non-real-time threads) from allocating CPU cycles and memory, and causes them to be blocked. Only critical real-time threads will continue to run. It is important to set the RTGCCriticalReservedBytes high enough to prevent a lack of free memory to run critical threads but not too high to prevent lower priority threads from running and causing a drop in throughput.

5.2 Experiments with RTS2.0

In this section, we describe the seven experiments performed with Java RTJ 2.0 beta release, with the RTGCCriticalReservedBytes set to 0. We set the RTGCCriticalReservedByte value to 0 so we can investigate the ideal configuration for the GIFC that achieves the maximum throughput. The new RTGC performed in a satisfactory manner and has met our expectations.

5.2.1 Experiments 1 to 4

These experiments were performed with Java RTS 1.0, and we described our findings in our previous technical report [6]. We ran them again using Java RTS 2.0 and did not observe any anomaly. In Experiment 1, a real-time thread, with the highest possible priority, creates a linked list of nodes. We confirmed that this thread does get preempted even if its priority is higher than the one for RTGC when the available memory is exhausted. Experiments No. 2 and No. 3 used NoHeapRealtimeThread objects, and as such, they are somewhat moot under RTS 2.0 since our recommended software architecture does not use no-heap real-time threads. In Experiment No. 4, we used only regular real-time threads (i.e., instances of the RealtimeThread class) dividing them into nominal and stateless discriminators (see Figure 8). If a stateless discriminator can finish its task within the designated deadline, the actual result is reported. If it cannot finish its task within the deadline, a default value (called the nominal result) is reported. We observed the expected behavior of getting more actual results when we increase the values for the deadlines.

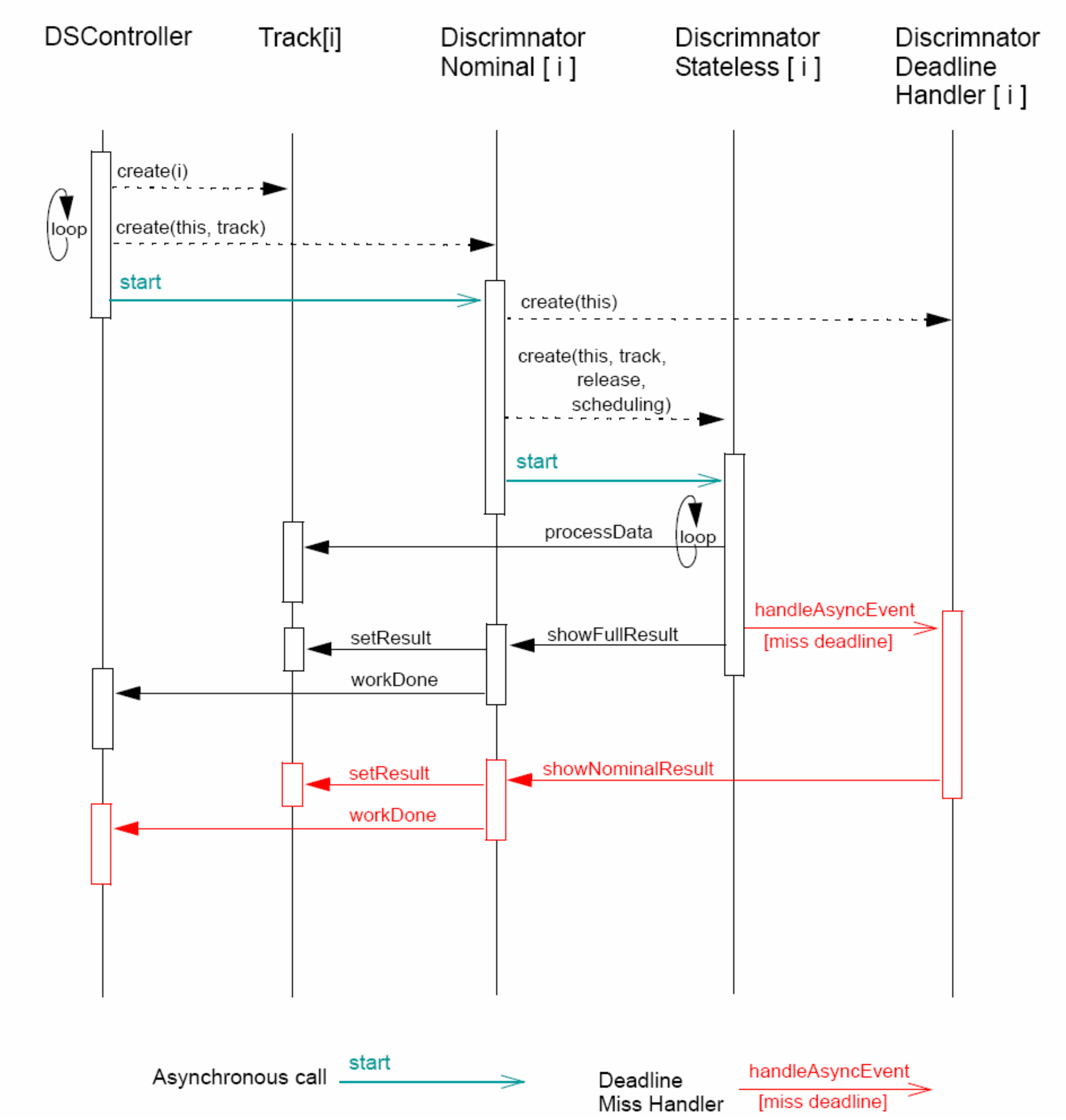


Figure 8. Sequence diagram for experiment 4

5.2.2 Experiment 5 - Interaction between RTGC and Real-Time Threads

The purpose of this experiment is to check the thread priority relationship between the real-time threads (instances of RealTimeThread and its subclasses) and the real-time garbage collector (RTGC). We define a descendant of RealtimeThread named RTGC_Tester. A RTGC_Tester simulates real-time computation by creating a linked list of N nodes with each node having an array of 500 BigInteger objects. By varying the values for N, we can study the impact of the real-time garbage collector to the running program when the garbage collection kicks in. The main driver creates and runs a number (M) of RTGC_Tester objects, where the value for M is an input to the program. RTGC_Tester objects are executed in sequence, and we track the elapsed time of each object in completing its execution. As more and more RTGC_Tester objects are executed, memory is consumed, and depending of the values of M and N, garbage collection would take place. We want to see how the real-time garbage collection would affect the execution of RTGC_Tester objects. We ran the test driver in two ways. In the first way, we set the priority of RTGC_Tester objects to the highest value and in the second way we set their priority to the lowest value. When there is no more free memory, the garbage collector will preempt any real-time thread (regardless of their priority) to reclaim memory from discarded RTGC_Tester objects. We observed this behavior in our test runs. The priority assigned to the RTGC_Tester objects is irrelevant when the free memory is exhausted. Results of the experiment with max priority and min priority RTGC tester are displayed in Tables 1 and 2, respectively. Notice Tables 1 and 2 display similar results.

Table 1: Experiment 5 Test Results with Max priority RTGC Tester

M (repeat count)	No of times GC occurred	No of spikes in elapsed time	Elapsed time of the interrupted RTGC_Tester	Comment
100	0	0	--	Elapsed time for the uninterrupeted RTGC_Tester is approximately 3 ms.
200	1	1	143 ms	
300	1	1	144 ms	
500	3	3	151 ms 141 ms 141 ms	
1000	6	6	174 ms 142 ms 138 ms 139 ms 137 ms 146 ms	

Table 2: Experiment 5 Test Results with Min priority RTGC Tester

M (repeat count)	No of times GC occurred	No of spikes in elapsed time	Elapsed time of the interrupted RTGC_Tester	Comment
100	0	0	--	Elapsed time for the uninterrupeted RTGC_Tester is approximately 3 ms.
200	1	1	144 ms	
300	1	1	144 ms	
500	3	3	144 ms 144 ms 140 ms	
1000	6	6	145 ms 141 ms 143 ms 142 ms 140 ms 140 ms	

5.2.3 Experiment 6 - Mixture of Critical and Non-Critical Real-Time Threads

In this experiment, we test a more realistic configuration where a mixture of critical and non-critical threads coexist in a running program. This configuration is closer to our proposed architecture of using high-priority Nominal and low-priority Stateless discriminators. For this experiment, we define two real-time thread classes: RTGC_Nominal and RTGC_Stateless. An instance of the RTGC_Nominal class simulates a Nominal object by spending time doing computation without allocating any memory in heap (uses only a local variable). An instance of the RTGC_Stateless class simulates a Stateless object by allocating a linked list of 2000 nodes with each node holding 500 BigInteger objects.

The main class of the experiment will create N RTGC_Nominal and N RTGC_Stateless threads in the initialization phase (where N is an input to the program) and then run the 2N threads concurrently. The RTGC_Nominal threads are run in the highest priority and the RTGC_Stateless in the lowest real-time thread priority. The priority of the RTGC is set to 40 as a runtime option. For each thread, we track its elapsed time.

Because the RTGC_Stateless threads allocate heap memory, we expect them to be interrupted and paused to wait for the garbage collector to complete its work, while the RTGC_Nominal threads sees no interruptions. The test runs confirmed our expectation. Table 3 shows some of the test results.

Table 3: Experiment 6 Test Results

N (repeat count)	No of times GC occurred	Minimum and Maximum elapsed times of RTGC_Nominal threads		Minimum and Maximum elapsed times of RTGC_Stateless Threads	
100	7	0 ms, 202417 ns	0 ms, 311083 ns	15 ms, 589583 ns	693 ms, 565833 ns
200	17	0 ms, 202333 ns	0 ms, 332916 ns	16 ms, 131334 ns	830 ms, 300750 ns
300	25	0 ms, 202750 ns	0 ms, 327750 ns	15 ms, 481501 ns	886 ms, 804750 ns
500	43	0 ms, 202500 ns	0 ms, 330583 ns	15 ms, 264834 ns	937 ms, 5332 ns
1000	90	0 ms, 201000 ns	0 ms, 455500 ns	15 ms, 188917 ns	882 ms, 228249 ns

5.2.4 Experiment 7 - Modified Experiment 4

In Experiment No 4, we used a deadline miss handler to process the timeout situation in which the DiscriminatorStateless is not able to compute the result within the allocated time period (see Figure 8). Because the deadline miss handler is associated to the DiscriminatorStateless that has a priority lower than the one for the RTGC, there is a possibillity of RTGC interrupting/delaying the asyncronous transfer of control to the deadline miss handler. To avoid this undesirable possibility, we used an alternative architecture to associate the deadline miss handler to a higher priority DiscriminatorNominal. The alternative architecture uses a timer, specifically, a OneShotTimer, to monitor the deadline for the stateless method. We designate the deadline to this OneShotTimer object by specifying a RelativeTime, such as 20 ms. When the set time is up, the OneShotTimer will trigger an event that allows the deadline handler to process the missed deadline.

We divide real-time threads into the two groups: those with a priority higher and those lower than the one for the RTGC. We call them the critical and noncritical threads, respectively. The key aspect of this architecture is that only the noncritical threads allocate memory in the heap. In our particular case, only the Stateless instances will allocate the heap memory. This architecture ensures that the critical threads will not get preempted by the RTGC, thus guaranteeing the determinism of the critical threads.

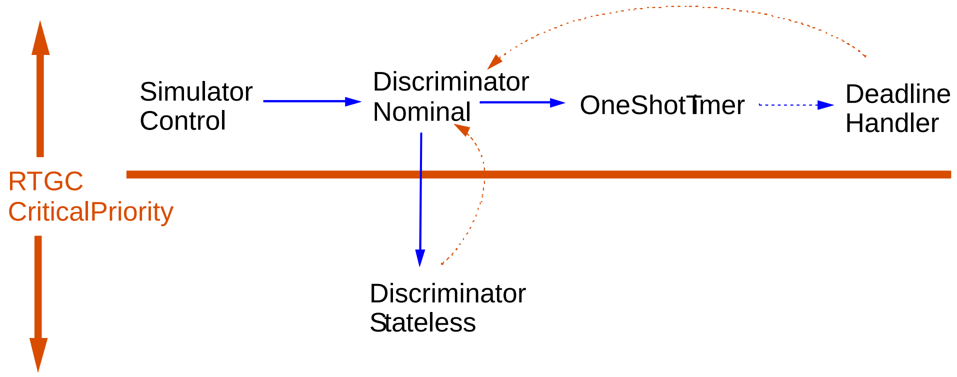


Figure 9: Collaboration diagram of the revised design

SimulatorControl is a RealtimeThread and its run method is defined as follows:

```
public void run( ) {

     for (int i = 0; i < N; i++) {
          Track node = new Track(i);
          DiscriminatorNominal disc
               = new DiscriminatorNominal(this, node);
          nominal[i] = disc;
          dicriminatorCnt++;
     }

     for (int i = 0; i < N) {
          nominal[i].start();
          /* A */
          /* Place delay here */
     }
}
```

At Point A in the code, after a nominal discriminator is started, we can place a time delay. Placing no delay means the program will run all nominal discriminators simultaneously. This could lead to an OutOfMemory exception when N becomes larger than a certain threshold. If we place certain amount of delay at Point A in the code, then it becomes possible for the nominal discriminators to complete its computation and call the SimulatorControl's workDone() method to turn themselves and memory allocated by the corresponding stateless discriminators into garbage for the RTGC to collect. A DiscriminatorNominal object performs the discrimination operation on a given track while a DiscriminatorStateless does the actual work of discrimination by interacting with its associated Track object. When the time duration set for the OneShotTimer is up, it calls its controlling DiscriminatorNominal to report that the nominal result must be used.

Table 4 summarizes the tests we ran using a varying number of discriminators, deadlines, and the pause time between the creation of discriminators (at point A in the code). The first tests used no delays (delay time = 0), the second had a 5 ms delay between the creation of discriminators and the third used 50 ms between the creation of discriminators.

Table 4: Results of Experiment 7

Deadline (ms)	N (# of discriminators)	Result with no delay (# of timeouts)	Result with 5ms delay (# of timeouts)	Result with 50ms delay (# of timeouts)
20	100	79 ~ 100	0	0
	200	200	0	0
	500	500	0	0
	1000	1000	0	0
	1500	OutOfMemory	OutOfMemory	1 ~ 5 (1 GC)
50	100	28 ~ 96	0	0
	200	142 ~ 200	0	0
	500	500	0	0
	1000	1000	0	0
	1500	OutOfMemory	OutOfMemory	0 (1 GC)
100	100	0 ~ 60	0	0
	200	35 ~ 200	0	0
	500	500	0	0
	1000	1000	0	0
	1500	OutOfMemory	OutOfMemory	0 (1 GC)
500	100	0	0	0
	200	0	0	0
	500	184 ~ 434	0	0
	1000	998 ~ 1000	0	0
	1500	OutOfMemory	OutOfMemory	0 (1 GC)

While increasing the deadline of the discriminators gives them more time to complete their computation and decreases the number of timeouts, we get an OutOfMemory exception for all values of deadlines when the number of discriminators, N is set to 1500 and no delay is used between discriminators. Even when we increased the delay time to 5 ms, we still get an OutOfMemory exception regardless of the values for the deadline when N = 1500, indicating that a 5 ms delay time is not long enough for the run method to be interrupted and the nominal discriminator to get a chance to call the workDone method. If we set the delay time to 50 ms we see that there is enough time for the RTGC to reclaim the heap memory in between successive runs of the DiscriminatorNominal threads.

6 Data Fusion and Tracking

A data fusion problem where a number of different types of sensors are deployed in the vicinity of a ballistic missile launch is studied [7]. An objective of this work is to calculate a *scoring function* for each sensor track, and the track file with the best (optimum) track score can then be used for guiding an interceptor to the threat within the boost phase. Seven active ground-based radars, two space-based passive infrared sensors and two active light detection and ranging (LIDAR) sensors are used to track the ballistic missile in the boost phase. Each space-based platform carries one passive infrared sensor and one LIDAR.

For the threat scenario, an IMPULSE© intercontinental ballistic missile model is used to create the trajectory of a generic ballistic threat. The IMPULSE model is developed by the National Air and Space Intelligence Center to provide an accurate representation of ballistic missiles [8]. Each sensor provides a track of the missile in the boost phase by using a multiple hypotheses tracking algorithm with an extended Kalman filter. The calculation of the track scoring function is to identify the sensor with the best track file. A track score is calculated for each sensor based on the kinematics of the missile flight parameters and the signal-to-noise ratio at the sensor. By using likelihood ratios, the optimum track file of the threat can then be determined and the corresponding track file can be transmitted to the battle manager control in order to lead the interceptor vehicle against the threat using the track file with the best score. Using the optimum track file scoring signal processing techniques developed in this work, the best track file can be sent to the interceptor to destroy the ballistic threat. This leads to a faster response management where the threat can be destroyed inside the territory of the country which launched the threat before any countermeasures are deployed.

The main goal was the evaluation of the performance of these sensors during the boost-phase stage of the missile's flight. The SNR of the sensors is used as a figure of merit. Seven X-band, ground based radars with the same parameters are deployed at different positions. Observing the results of the radars, it is inferred that the distance between the radar and the target is the basic factor to be taken into account for evaluating the performance of the sensor in the data fusion process. Two IR sensors are deployed in two LEO satellites in order to evaluate their performance during boost-phase. They have an advantage over radars due to their capability to detect the radiation of the missile in this phase. Two LIDAR sensors are also deployed on the same platforms along with the IR sensors. The LIDAR sensor is used to provide the third dimension, the range, which an IR sensor cannot calculate. Based on the results, it is inferred that the use of the LEO satellite supported by IR sensors improves the detection and tracking performance of a ballistic target in the first stage of its flight. The drawback is that the area covered by a LEO satellite for surveillance is limited, so a network of LEO satellites is required for covering the Earth. The next goal of the work was the fusion of the radar sensor data. The technique of likelihood ratios for tracking of each sensor is used. The evaluation provides results about the performance of each sensor with respect to the scan number, which in this work represents the time of the flight.

7 Ballistic Missile Simulation and Tracking

This research extends the tracking and backfitting techniques used earlier by the Naval Postgraduate School [9-13] and compares Kalman-based and particle filter-based algorithms with hyperbolic backfit formulas and genetic algorithms. The IMM filter introduced here dates back to work by Blom [14], and it has been modified and adapted for tracking by Blom, Bar-Shalom and others [15][16].

Using three dimensional measurements (range, azimuth and elevation) available to the tracking algorithms we created a simulation for the trajectory of a ballistic missile. The missile was modeled to have an 8 g acceleration equivalent thrust. Gravity acceleration was assumed to be constant and atmospheric drag used an exponential model for atmospheric attenuation with altitude[17]. The acceleration vector used in the missile simulation was:

$$a = T - 0.001 \times \sigma(z) \times \|v\|^2$$

where $\sigma(z)$ is the height-dependent atmospheric density, v is the missile speed, and T is the thrust. This acceleration vector is implemented in the direction of the velocity vector to emulate a gravity turn.

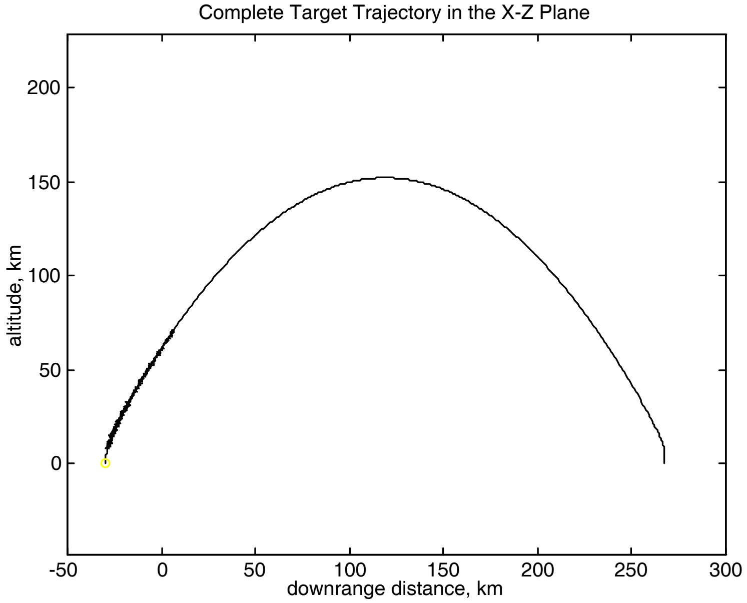


Figure 10: Ballistic missile trajectory

The missile was launched at a vertical angle of 87.5 degrees and performed a gravity turn in the (x,z) plane, z being altitude. The launch point (x,y,z) was (-30000, 40000, 0) and the sensor was at (0,0,0). No noise was added to the simulated missile dynamics. Measurement errors were taken as 10 meters standard deviation in range, 0.7 degree in azimuth, and 0.7 degree in elevation. Measurements were mapped into Cartesian coordinates, and the measurement equation was assumed linear in both the Kalman and the particle filter. Measurements were assumed to occur every 0.1 second.

7.1 Results of Simulation Comparison

The results of the simulation demonstrated that the IMM performance is superior to the Kalman filter with acceleration estimates, particularly during the coast phase of the missile.

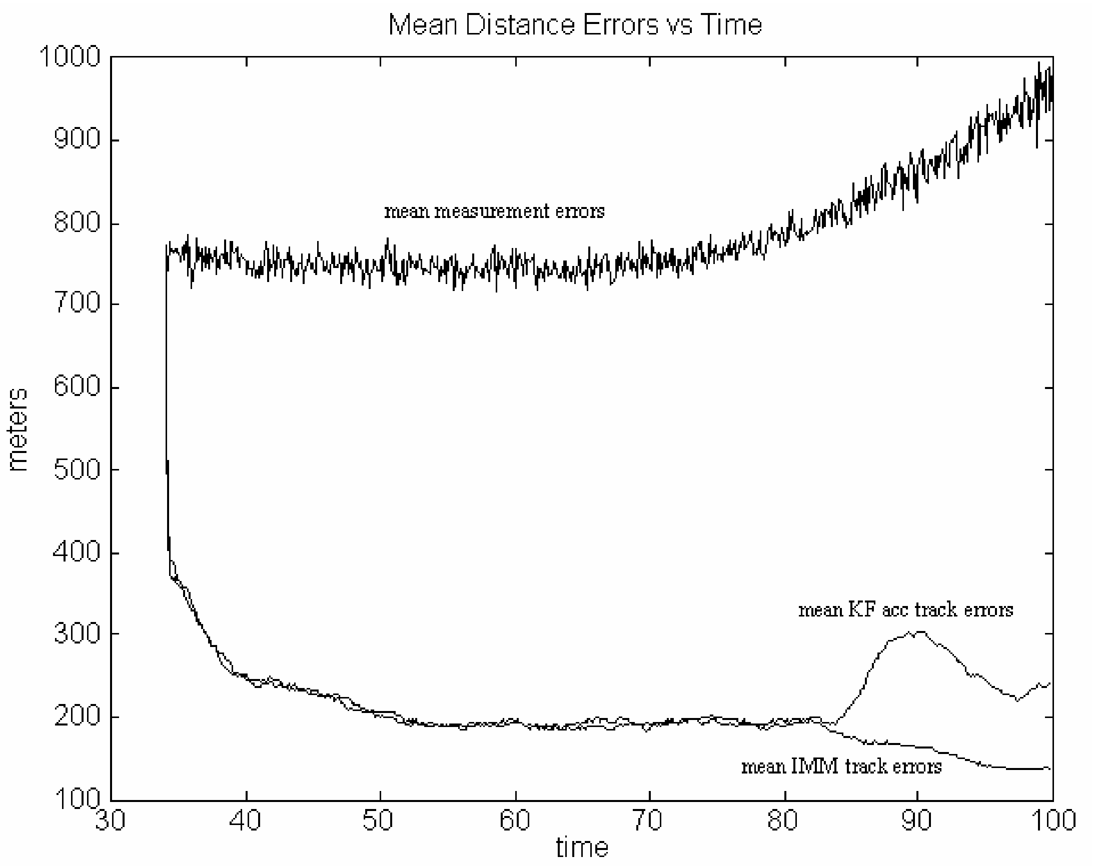


Figure 11: Mean distance errors: IMM and KF acc track comparison

Further simulations of the Kalman backfit estimate for launch point determination demonstrated a bias. Depicted in Figure 12 is the 95% error ellipse generated by the 1000 data point sample covariance.

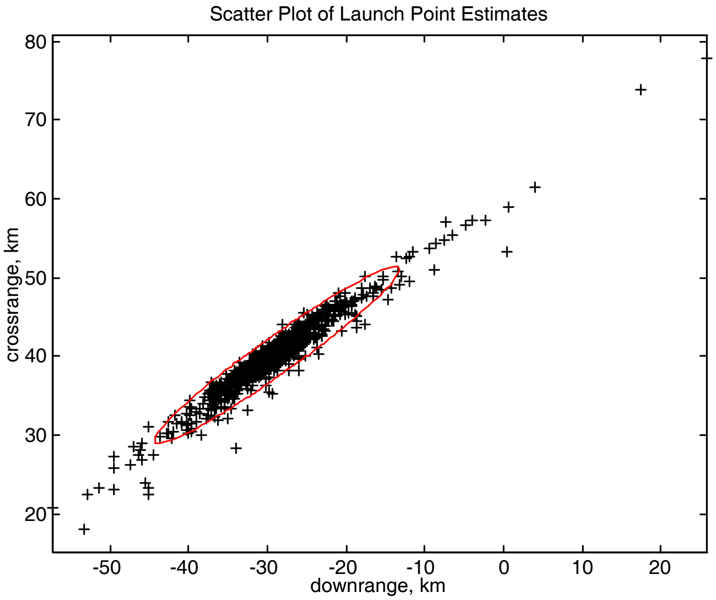


Figure 12: Scatter plot of launch point estimates with 95% ellipse

The inner error ellipse depicted in Figure 13 is a 95% error ellipse for the mean, and the mean of the data sample is at the center. The actual launch point lies at the point depicted in the plot, noticeably outside the mean error ellipse, indicating this procedure tends to produce a biased estimate. The error in the mean estimate is over 1.2 kilometers. The maximum distance from the sample mean to a point on the inner ellipse is approximately 600 meters.

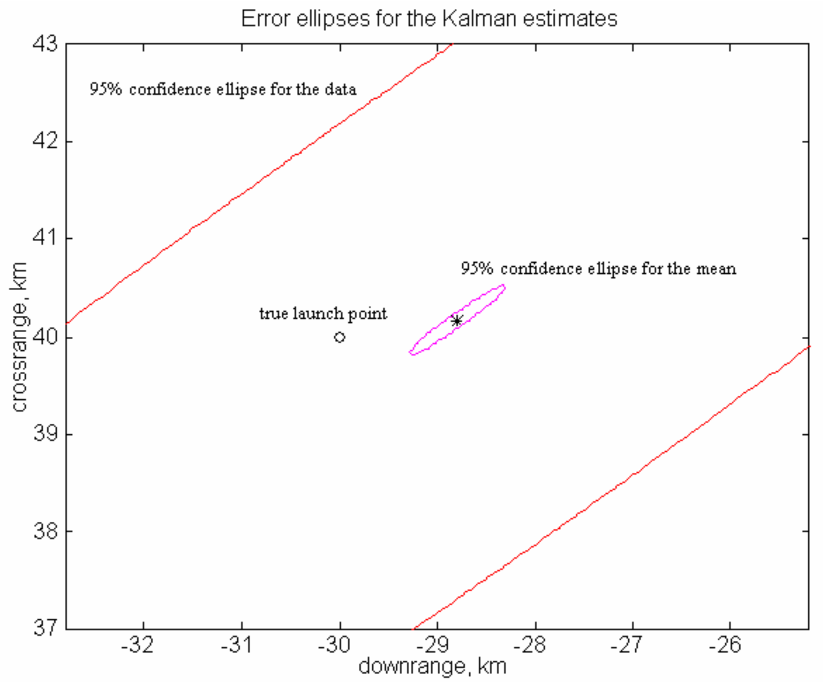


Figure 13: Confidence ellipse excluding true launch point

An analysis of individual simulations indicated that the largest source of error in the Kalman backfit algorithm occurred when the backfit trajectory did not reach the ground (indicating the error in the observations was leading to velocity estimates that were too small for the altitude at which the observations occurred).

The Hyperbolic Backfit, designed to fit the observation data to a hyperbola with the origin on the ground, was found to run quite fast and have a better performance than the Kalman Backfit with the actual launch site inside the confidence region of the mean.

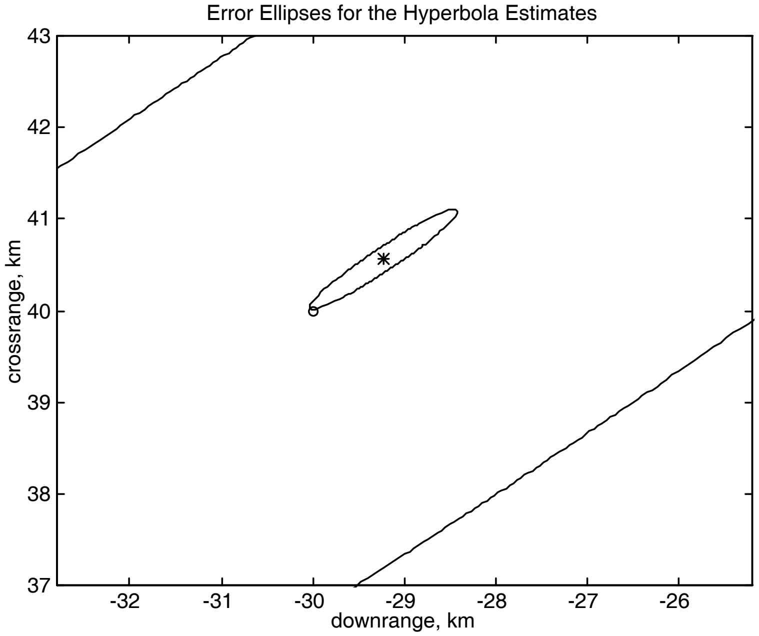


Figure 14: Hyperbola launch point estimation mean

A particle filter-based algorithm and a genetic algorithm for estimating missile launch point were also investigated using measurement data generated as described above for the Kalman and hyperbolic procedures. The genetic algorithm produced results comparable to the Kalman backfit, but ran an order of magnitude slower. The particle filter ran extremely slow, over three orders of magnitude slower than the Kalman and Hyperbolic backfit algorithms, with no discernible improvement in estimation error. In fact, for the few cases completed, the results were worse than the Kalman backfit algorithm. From these results it appears that neither of these algorithms would be adequate for use in a real-time system.

8 Multiple Hypothesis Tracking [18]

In this research, we investigate the use of the multiple hypotheses tracking (MHT) algorithm to process track files from a single sensor of a surface-based sensor network. In particular, a framework is developed for an efficient form of the MHT, specifically, the linear assignment problem approach as developed in [19] is included and is used to quickly solve for the N-best association hypotheses and, thus, expediting the tracking process. This work applies the algorithm to the simulated multiple-ballistic missile launch as described above and examines the feasibility and appropriateness of the modified algorithm for this specific application. More importantly, a study of the algorithm's performance is made and we discuss its computational complexity in this particular setting.

8.1 Functional Areas

There are several functional areas considered in this work: the missile threat profile, the sensor model, and the tracking algorithm. Figure 15 shows the main blocks considered throughout the remainder of this paper. The missile flight data is generated with the IMPULSE© simulation toolkit (left-most block). Realistic test data is generated to qualify our implementation of the multiple hypotheses tracking algorithm

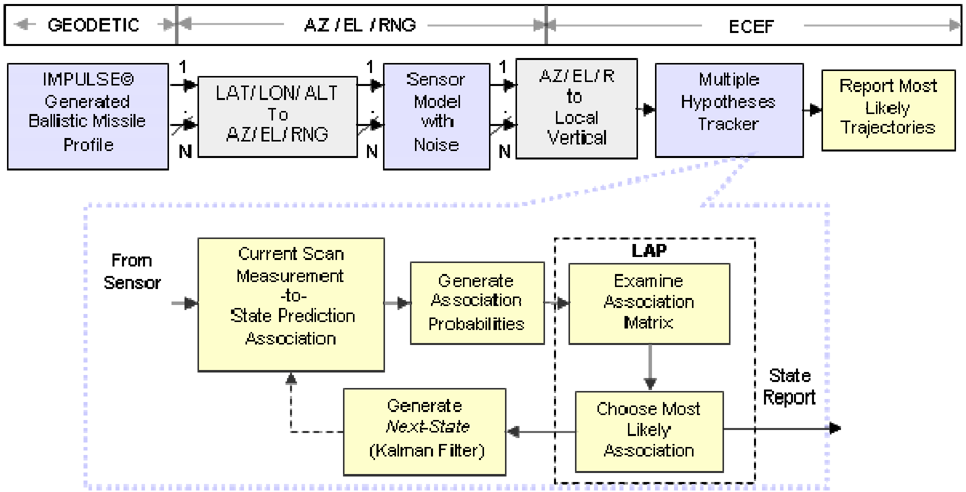

GEODETIC
AZ / EL / RNG
ECEF
IMPULSE© Generated Ballistic Missile Profile
1
N
LAT/ LON/ ALT To AZ/ EL/ RNG
1
N
Sensor Model with Noise
1
N
AZ/ EL/ R to Local Vertical
Multiple Hypotheses Tracker
Report Most Likely Trajectories
From Sensor
Current Scan Measurement -to- State Prediction Association
Generate Association Probabilities
LAP
Examine Association Matrix
Choose Most Likely Association
Generate Next-State (Kalman Filter)
State Report

Figure 15: Main functional areas implemented in this study to include the IMPULSE© threat profile, the RF sensor model, and the multiple hypothesis tracking algorithm

8.1.1 Sensor Modeling

The second key functional area in this work is the simulation of the radio frequency (RF) sensor that is used to "detect" the missile launches and provide position measurement updates over time. An X-Band, low pulse repetition frequency (LPRF), monostatic radar is considered and is used to generate observations of the missile fly-out. The signal-to-noise ratio (SNR) of the sensor is used to introduce precision error into the sensor-to-missile observations. The SNR (S/N) for a single-pulse radar is given as [20]

$$S/N = \frac{P_t G_t^2 \tau \lambda^2 \sigma}{(4\pi)^3 k T_0 F R^4}$$

where P_t is the peak transmitted power, G is the antenna gain, σ is the radar cross section of the object , τ is the compressed pulse width, λ is the wavelength, F is the noise factor, k is Boltzmann's constant, T_0 is the system temperature, and R is the range to the target. The SNR quantity obtained from the above equation is used to calculate the range precision error and the angular precision error, as given by the following equations

$$\sigma_R = \frac{c}{2B} \quad \frac{1}{K\sqrt{(2(S/N)n_p)}} \quad \textbf{and} \quad \sigma_\theta = \frac{\theta_{3dB}}{K\sqrt{(2(S/N)n_p)}}$$

where $B = 1/\tau_c$ (or the inverse of the sub-pulse width), c is the speed of light, n_p is the number of integrated pulses, θ_{3dB} is the half-power beam-width, and $1 \le K \le 2$ is an error-slope coefficient. In this study, K assumed the value 1.7 for a monopulse radar.

8.2 Multiple Hypothesis Tracking – Linear Assignment Approach

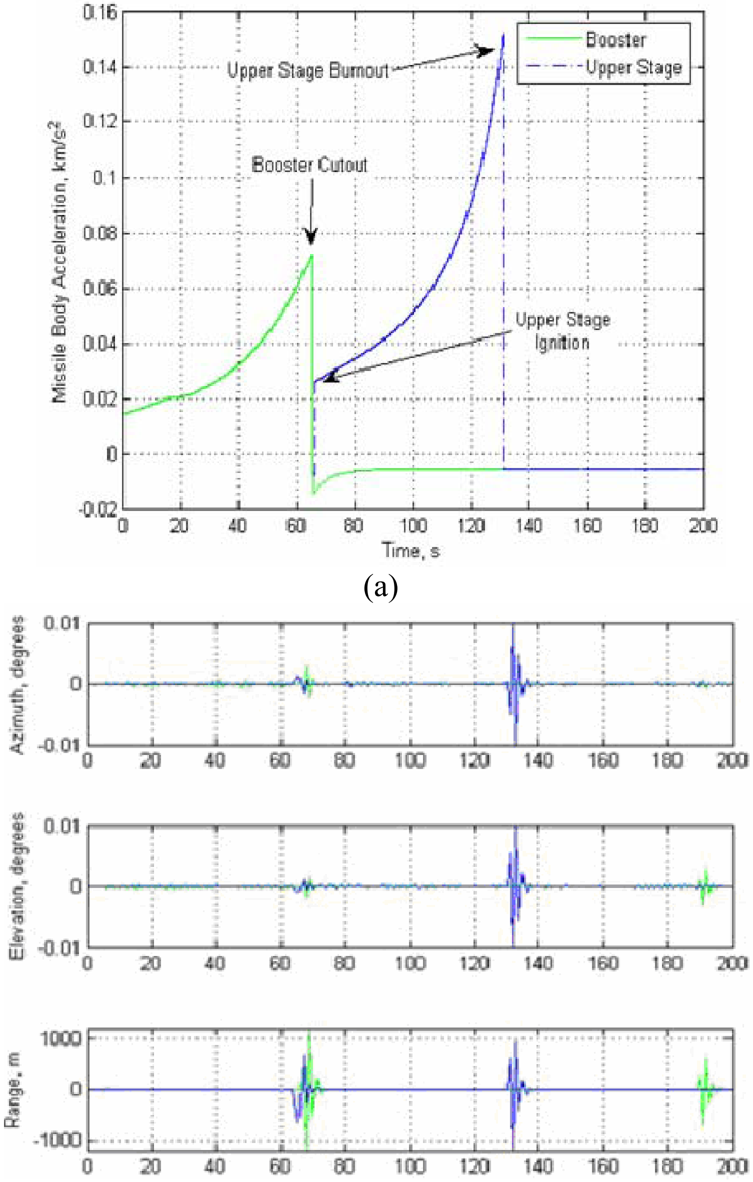
In a multiple target environment, the primary difficulties correctly assigning successive contacts to each corresponding established target's next-state prediction. This process of making possible assignments is referred to as measurement-to-target association. An association hypothesis maps each measurement to possible target next-state predictions. Furthermore, there may be hundreds of such potential hypotheses due to the unpredictable nature of each target as they move in a multiple-target environment. Consequently, the number of possible associations that must be examined is enormous.

A critical drawback of the MHT, as developed by Reid and later modified by Nagarajan, Chidambara, and Sharma in [21], is its overall growth in computational requirements. As the number of targets in a scanning region increases, the number of measurement-to-known-targets hypotheses increases exponentially; thus, the standard approach to the MHT is not practical when processing power and memory are limited. A modification on the MHT, namely the inclusion of the *linear assignment problem* (LAP), provides an efficient means of identifying the likely measurement-to-target associations. The method for determining the association likelihood probability as outlined in [19] is used in this work and serves as an efficient means to successfully identify correct target-to-next-measurement pairings.

8.3 Experimental Results

The MHT with the LAP method was applied to the IMPULSE-generated ballistic missile launch scenario. Recall that IMPULSE© provides realistic motion modeling of missile flight as it incorporates many physical parameters when generating a flight profile. The modeling tool enables the study to use missile data that exhibit staging—jettisoning of spent components—so we may study the success of the algorithm as the number of measurements increase. The staging events also present challenges for tracking algorithm as the sudden loss of thrust and re-thrust at booster jettison introduces some nonlinear acceleration gradients. The acceleration profile and effect on the innovation can be seen in Figure 16. Despite the non-linear accelerations due to missile staging and introduction of new targets—the spent booster cans—in the sensor's view, the MHT is successful in tracking each missile body. This can be seen in Figure 17(a).

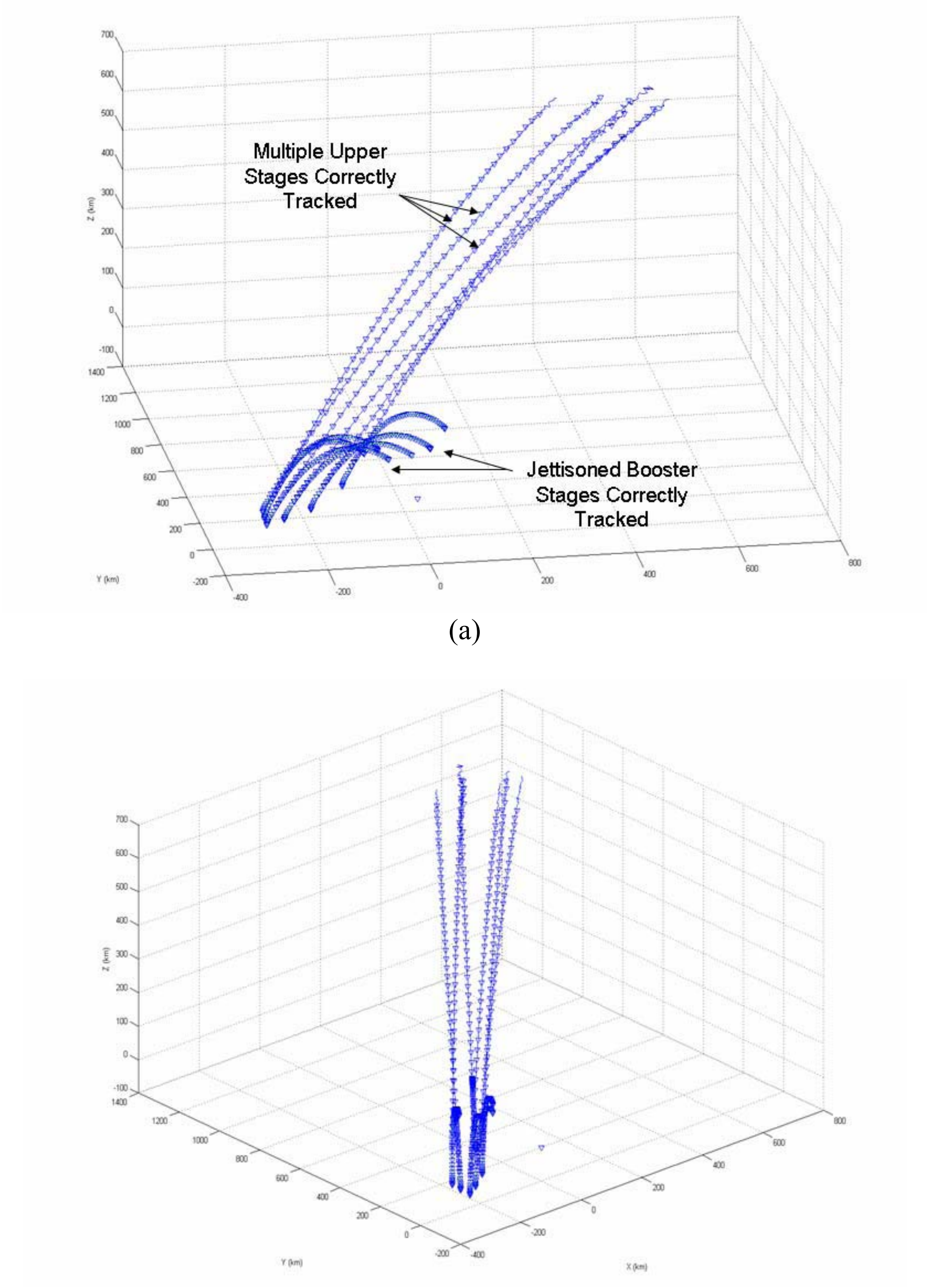
The algorithm also tracks each flight trajectory even though they may cross paths in the sensor's view as seen in Figure 17(b). The viewpoint orientation is from behind the missile launch (missiles travel away from observer) and emphasize the tracking of crossing flight trajectories. The tracking problem in this study examines at most 12 targets throughout the entire simulation, thus, for this experiment; there are 479 million potential hypotheses (M!) per time scan. However, the linear assignment problem is still computationally effective and efficient. The key advantage to linear assignment approach is that only a limited number of sweeps must be conducted to find the N-best solution.



(a)

(b)

Figure 16:(a) Missile acceleration; peaks are indicative of staging. (b) Innovation, $\widetilde{Z}_{m,j}$, plot as reported by extended Kalman filter within the MHT. Peaks coincide with staging.



(a)

(b)

Figure 17: Missile trajectories as tracked by the MHT for (a) South-to-North view, (b) Easterly view to emphasize correct tracking through missile flight path crossing.

9 Distributed Medium Access and Data Dissemination for Cooperative Radar Systems

In this section, we discuss research efforts that propose efficient and effective medium access and data dissemination schemes for a distrubted set of radars employed in BMDS. Both of these algorithms view the wireless network of radars as a large-scale wireless sensor network and leverage ongoing research in sensor networks.

9.1 CR-MAC: A Distributed Medium Access Control Protocol for Wireless Networks of Cooperative Radar Systems [22]

In this research, the medium access control issues of a distributed wireless network of cooperative tracking radars are investigated by modeling the individual radars as part of a system-of-systems rather than as independently operating sensors. Two fundamental observations are made regarding the radar control traffic flow and the radar data traffic flow: the pulse repetition frequency (PRF) is found to be bounded by the radar control packet delay and the radar data traffic is bursty but not well-suited to contention-based medium access. Motivated by these observations, we propose a novel medium access protocol: the Cooperative Radar Medium Access Control (CR-MAC) protocol. CR-MAC is an application-aware protocol that combines the throughput of a time division medium access (TDMA) protocol with reduced delay of a contention-based protocol. Control packet delay as well as data traffic throughput are analyzed and CR-MAC is found to outperform conventional TDMA and CSMA.

CR-MAC is a cross-layer, "application-aware" protocol that provides high data throughput through the use of a time division multiple access (TDMA)-based contention-free architecture overlaid with a contention-based priority mechanism designed to accommodate the delay-sensitive control packets. This protocol provides medium access for both the data and control packets in a single channel.

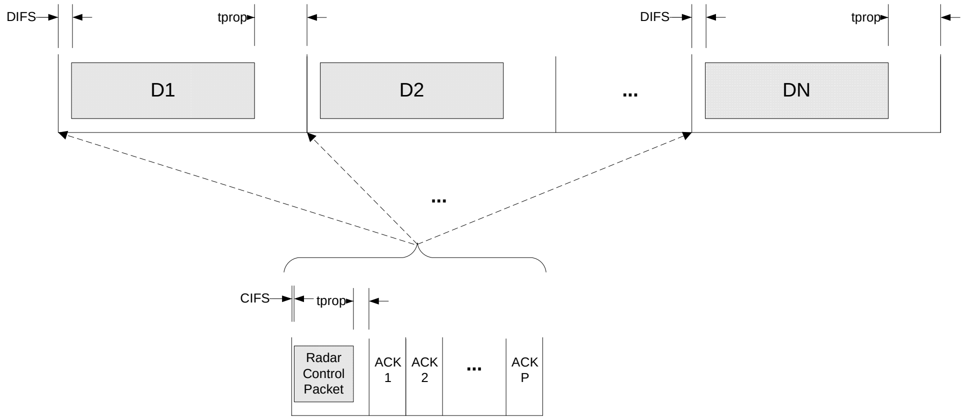


Figure 18: CR-MAC frame structure

We analyze the radar control packet delay as well as the data packet throughput and compare the performance of CR-MAC to conventional TDMA and CSMA schemes. Figures 19 and 20 show several results from the comparison. It can be seen that the throughput for CR-MAC does not fall off until PRF exceeds 100,000 pulses/sec.

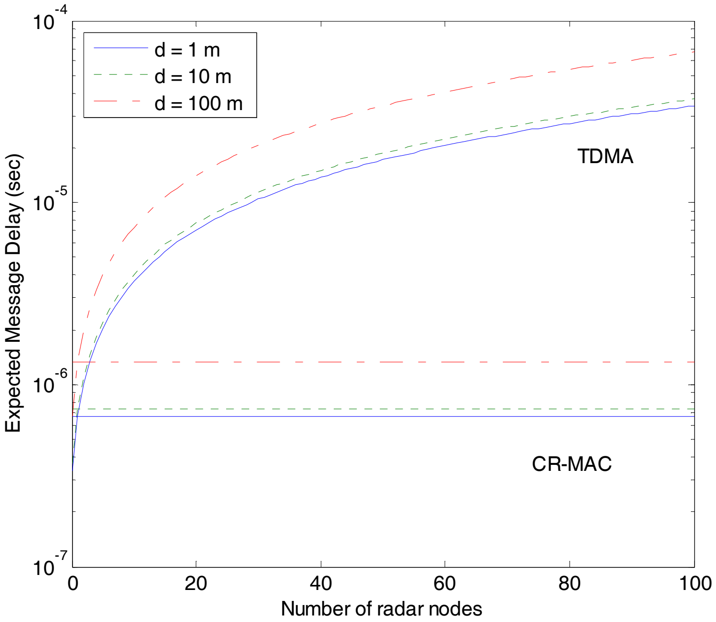


Figure 19: Maximum control packet delay for TDMA vs. CR-MAC

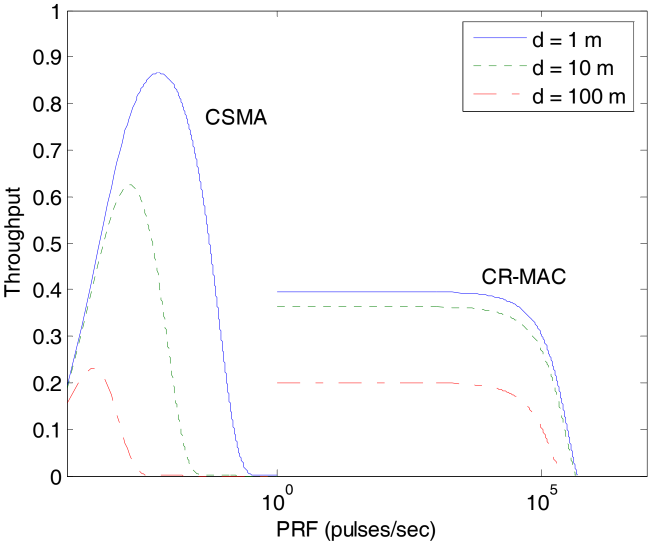


Figure 20: Data Traffic Throughput vs. PRF for CR-MAC vs. CSMA

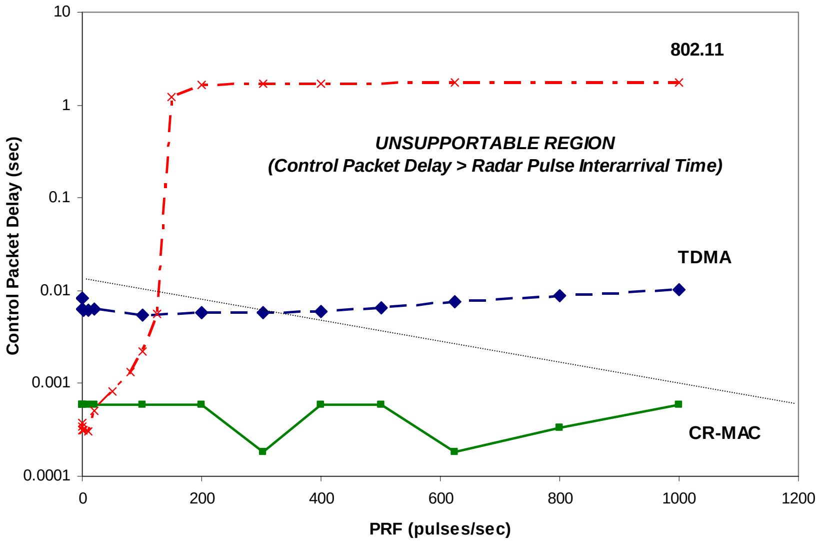


Figure 21: Mean Control Packet Delay vs. PRF for CR-MAC, TDMA, and 802.11

The performance of CR-MAC was compared to that of conventional TDMA and 802.11 through simulation using OPNET (see Figure 21). As expected, CR-MAC is found to outperform TDMA by approximately an order of magnitude. This performance margin is proportional to the number of active nodes. The CMSA/CA scheme of 802.11 performs well at low loads, but delay rapidly increases when the network becomes saturated as the control packets contest for the medium with the increasing load of data packets.

9.2 Data Dissemination Algorithm for a Hybrid Large-Scale Wireless Sensor Network [23]

In this research, a hybrid, large-scale wireless sensor network (WSN) for missile defense is introduced that consists of terrestrial and satellite nodes. Rather than treating each node as an individual system, we model this sensor network as a system-of-systems with the objective of real-time tracking of multiple ballistic missile threats. We present a hybrid data dissemination algorithm based on a combination of data centric routing concepts and clustering mechanisms. Through simulation, it is shown that the performance of this hybrid algorithm significantly improves data throughput without sacrificing the real-time data delivery requirements associated with missile tracking applications.

The terrestrial nodes will be both mobile and fixed RF and IR sensor platforms, such as land-based station, warships, fixed-wing aircraft, and unmanned air vehicles (UAV). The satellite sensor platforms will be Geostationary Earth Orbit (GEO) and Low Earth Orbit (LEO) satellites, carrying search and track IR sensors, respectively.

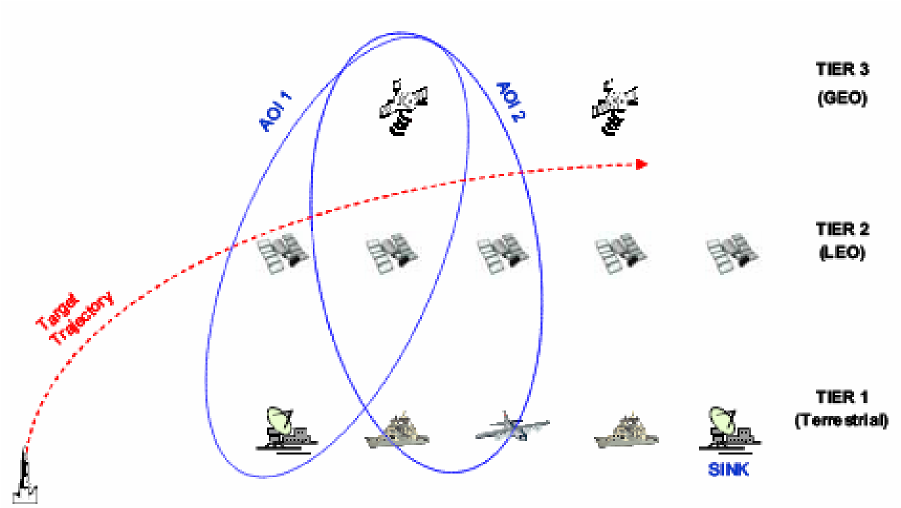


Figure 22: Hybrid Wireless Sensor Network and Area of Interest (AOI)

Considering the unique characteristic of a network designed to support missile defense, we propose a data dissemination algorithm that combines the approach of data centric routing protocols with the use of a clustering mechanism called the "Area of Interest" (AOI) (see Figure 22). At any given point in time, this AOI will include a GEO satellite and the sensor nodes that are currently tracking the target (LEO satellite and/or terrestrial nodes). We model three major data flows in our network: (1) a flow among the sensor nodes that perform target tracking, (2) a backflow of target data traffic towards the sink, and (3) an engagement order flow from the sink to the weapon platforms involved in the target interception. For the backflow to the sink, we introduce data aggregation within the current AOI. The proposed data dissemination algorithm is shown in Figure 23.

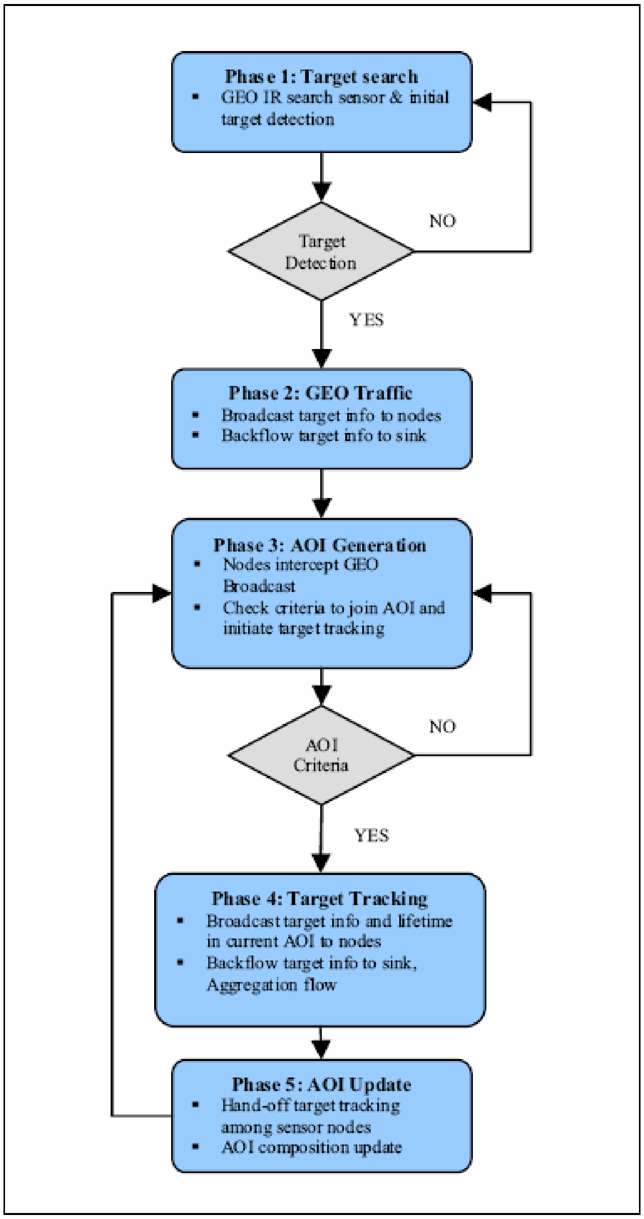

Phase 1: Target search
GEO IR search sensor & initial target detection
Target Detection
NO
YES
Phase 2: GEO Traffic
Broadcast target info to nodes
Backflow target info to sink
Phase 3: AOI Generation
Nodes intercept GEO Broadcast
Check criteria to join AOI and initiate target tracking
AOI Criteria
NO
YES
Phase 4: Target Tracking
Broadcast target info and lifetime in current AOI to nodes
Backflow target info to sink, Aggregation flow
Phase 5: AOI Update
Hand-off target tracking among sensor nodes
AOI composition update

Figure 23: Flowchart of the proposed data dissemination algorithm

The simulation results shown in Figures 24 and 25 evaluate the composition as well as the throughput and delay performance of the AOI mechanism. It can be clearly seen that aggregation improves throughput across the full range of network load

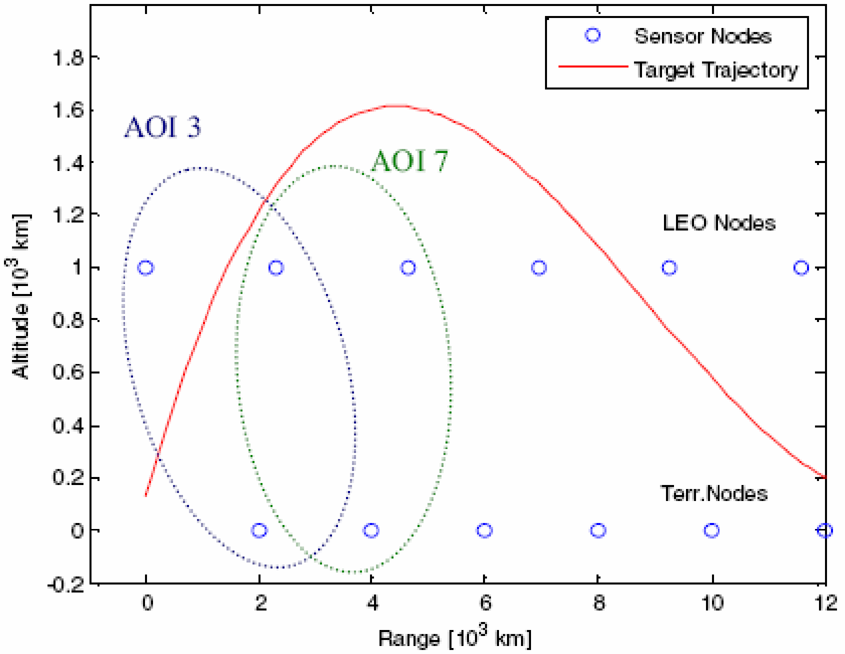


Figure 24: AOI composition and target track

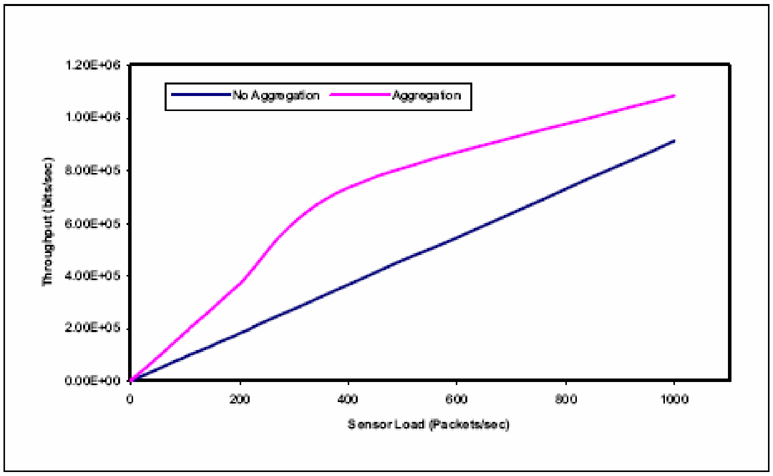


Figure 25: Network Throughput

The improvement in throughput with data aggregation in the AOI comes at a tradeoff of increased end-to-end delay as can be seen in Table 5.

Table 5: Overall end-to-end delay

Scenario	Delay (msec)
No Aggregation	31
Aggregation	36

9.3 Multistage Network Security [24]

We also proposed an efficient multistage security mechanism for node and data authentication and data confidentiality. Node authentication is provided by digital signatures and the public key infrastructure (PKI). The TESLA algorithm and IPSec are utilized for data authentication and confidentiality, respectively. Performance analysis and simulation results demonstrate that the proposed mechanism meets the real-time data dissemination requirements of a ballistic missile defense system while maintaining throughput commensurate with unencrypted Internet Protocol (IP).

9.4 Sensor Motes for Target Tracking

We conducted a simple target tracking experiment using a network of Crossbow sensor motes. In the experiment, the onboard acoustic sensors (microphones) emulated the RF/IR sensors of a ballistic missile defense network. Each sensor node contained a transceiver operating at 2.4 GHz for communication and an acoustic sensor. The acoustic sensors used have a range of 2-3 feet, and their accuracy rapidly degrades as the distance between the sensor and the target increases. In the first step of the experiment, the sensitivity and range of coverage of the acoustic sensors was determined. Followed by that, a network of nine of these sensor nodes was formed to conduct the tracking of a moving object through the sensor field. An object with an onboard acoustic source is then maneuvered through the sensor field, and the measurements at each of the nine sensors are recorded. By combining the measurements, we found that the object's location within the sensor field could be determined with an accuracy of 5% or greater. These are very preliminary results and the techniques used to combine the measurements are simplistic. Nevertheless, this effort could be expanded to implement and test the algorithms developed and tested in Matlab for tracking and fusion.

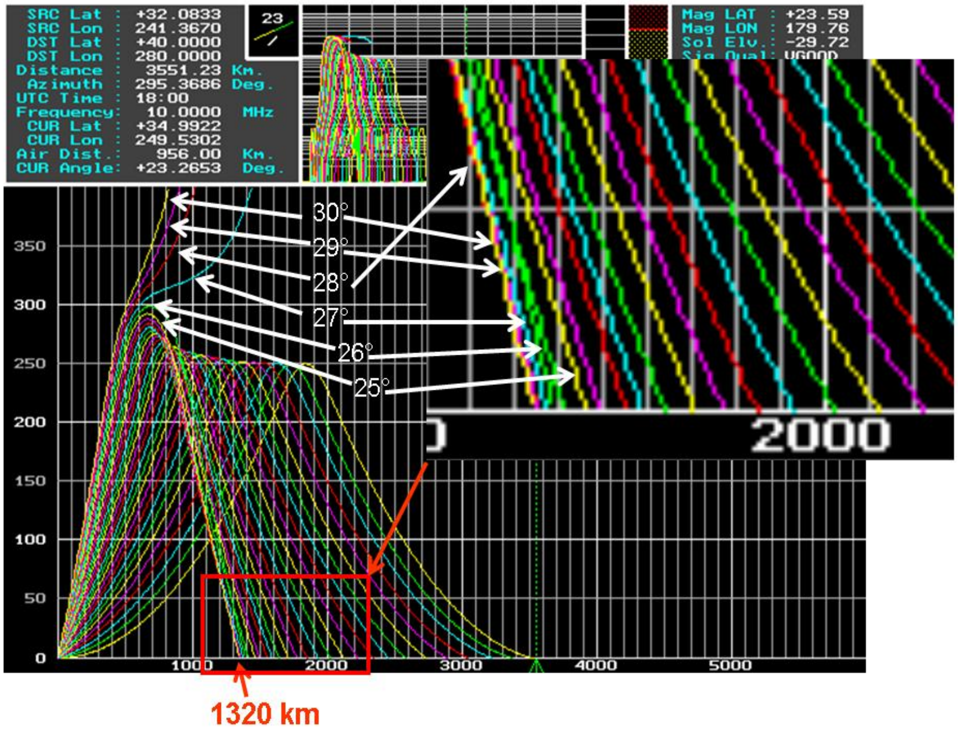
10 Missile Defense Technologies

This section will cover several technologies and methods being used or developed that can be employed in a BMDS. These technologies and methods provide a solution to tracking and intercepting a long range ballistic missile.

10.1 Predicting Threat OTH Radar Footprints [25]

MDA planners are preparing to defend the U.S., their outposts and those of our allies against theater and intercontinental ballistic missiles from industrialized foes. Over the Horizon (OTH) radar systems plays an important role in targeting these threat missiles. China is enhancing its battlefield surveillance ability especially within the Taiwan Strait. The acquisition of modern intelligence, surveillance, and reconnaissance (ISR) systems remains a critical aspect of Beijing's military modernization. China is developing its ISR capabilities based upon indigenous developments, supplemented by foreign technology acquisition, and procurement of complete foreign systems. Its capabilities can be enhanced by new space systems, airborne early warning aircraft, long range unmanned aerial vehicles, and OTH radar. There are as many as three Chinese OTH sky-wave radar systems used to target aircraft carriers and provide an early warning capability. Also, there is at least one surface-wave OTH radar in mainland China. OTH radar enhances China’s ability to detect, monitor and target naval forces in the Western Pacific Ocean and the Taiwan Strait [26][27].

PROPLAB ray tracing simulation software was used to predict the path of the HF signal from a Chinese OTH radar system propagating in the ionosphere. The location chosen for simulation of the OTH radar system is Nanjing, China. The results from the simulations revealed the properties of reading OTH radar. Figure 26 shows the graph of one of our experiments. It shows the ray tracing using only one hop and we can observe that the skip distance is about 1320 km which is also where the distance changes direction (from decreasing to increasing) at the angle of 28 degrees.



```
SRC Lat   : +32.0833
SRC Lon   : 241.3670
DST Lat   : +40.0000
DST Lon   : 280.0000
Distance  :  3551.23 Km.
Azimuth   : 295.3686 Deg.
UTC Time  : 18:00
Frequency:  10.0000  MHz
CUR Lat   : +34.9922
CUR Lon   : 249.5302
Air Dist.:   956.00  Km.
CUR Angle: +23.2653  Deg.
```

```
Mag LAT  : +23.59
Mag LON  : 179.76
Sol Elv.: -29.72
Sig Qual: UGOOD
```

Figure 26: One-hop ray tracing with different degrees

10.2 Electronic Warfare Metrics

The technical concept of this work is to continue the development of our network-centric EW metric simulation to assess the value of information and networking when active RF sensors and communication links are used to support the MDA Advance Battle

Manager. The simulation (using MATLAB) provides a comprehensive analysis of network capabilities, and includes the different wireless technologies and topologies used for electronic warfare [28][29][30]. This project will also include a detailed model to predict the important network EW metrics such as connectivity measure, reference connectivity measure, network reach, and network richness. Also quantified will be the characteristic tempo of various network connections and the impact on the command and control. The simulation will use the EW metrics to quantify the operational tempo of the OODA (observe, orient, decide, act) loop during a ballistic missile launch.

When analyzing a network approach to EW, the typical approach is usually to assume that there are a large (asymptotic) number of nodes that are homogeneous and the network does not evolve in time during combat. The role of information in combat however, is nonlinear and a complex problem [31]. The new capability of our proposed simulation approach is that it quantifies the network centric electronic warfare *metrics* for any multi-faceted EW problem involving technology, organization structure, command and control and human factors (such as cognition and decision making).

The metrics that are quantified with our simulation will capture the degree of networking, the network topology and communication modes. It will also quantify the level of shared awareness and the quality of decisions that are made using the information provided by the network. Since the capability value of each node is separately modeled, the re-routing option impact can be easily examined. In addition, the performance of a real network under electronic attack can be analyzed easily.

The rate of change as a function of the number of links that are cut, nodes being degraded (or attacked) provide insight into the network robustness. Another novel aspect of our simulation is quantifying the richness of the network which is a measure of the average rate that information entropy (or knowledge) is generated through the network. With the network richness and network reach, the characteristic tempo can be quantified. The characteristic tempo reflects the network's finite maximum rate of information exchange which is limited by the technologies employed and the network topology. The speed of the C2 and the action tempos involved will also be incorporated in the simulation to quantify the operational tempo of the OODA (observe, orient, decide, act) loop. In summary, with this simulation we are able to realistically assess the effectiveness of EW employment for any general type of network scenario.

11 Attack Laser ABM

The Air Force YAL-1A Attack Laser, a.k.a. Airborne Laser (ABL), is a component of the BMDS. The fundamental objective of the BMDS is to develop the capability to defend United States forces, territories and allies against all classes of ballistic missile threats. The BMDS suite will include other complementary missile interceptors, land, air, sea and space-based sensors, and battle management command and control systems. The MDA will develop, test, and continuously evaluates operational alternatives for "hit-to-kill" technology, which the agency describes as "hitting a bullet with a bullet" [32].

MDA is developing BMDS technology for engaging missiles in three phases of launch – the boost phase, the midcourse phase, and the terminal phase. The boost phase is the phase in which the missile's warhead is still attached to the booster rockets. In this phase, the missile is more vulnerable to counterattack because of the fuel it contains. Countering missiles in this phase is also ideal because it results in the missile falling to the ground nearer the adversary's territory. The ABL is one system that the MDA is fielding to counter ballistic missiles in the boost phase. The ABL mission is to find, track, engage and destroy small, short-range surface-launched missiles while they are still in the highly vulnerable boost phase of flight. Operating above the tropopause, the system will autonomously detect and track missiles as they are launched, using an onboard surveillance system. The system will acquire the target, then accurately point and fire a directed energy laser to sufficiently weaken the missile casing during flight to cause structural failure. The directed energy laser has a revolutionary adaptive optical system capable of focusing extreme heat onto a basketball-sized spot from hundreds of miles away. The ABL can also transfer information on launch sites, targeting, tracking and predicted impact point information to other components of the BMDS.

The ABL's megawatt-class kill laser is one of the oldest technologies on the ABL platform. It was developed by the Air Force in the 1970s, the laser functions by a chemical reaction between chlorine, hydrogen peroxide and iodine to create an explosion of light. This light travels down a mirrored tube and flexible hose to the rotating nose turret. The ABL is expected to be able to fire 20 to 40 kill shots before requiring landing for maintenance. In addition to missile defense, the ABL will have inherent capabilities to perform other activities, such as engaging threat aircraft, temporarily blinding enemy satellites, performing imaging surveillance and providing cruise missile defense.

To date, all components of the ABL have been successfully tested individually, and the MDA is currently conducting series of tests with the components integrated as a complete system. When complete, the ABL will meet an urgent requirement that has been around since DESERT STORM, since Iraqi Scuds were used to bomb buildings and kill our troops. This capability has been in development by the Air Force since 1993, but long before then, extensive developments occurred that set the stage for today's ABL project.

The ABL successfully demonstrated its laser targeting system against an airborne target for the first time in March 2007 [33]. The target was a white, missile-shaped silhouette painted onto an NC-135E. After taking off from Edwards Air Force Base, California, the ABL aircraft used its infrared sensors to find the simulated missile exhaust, then pointed and fired its tracking laser at the target, and successfully calculate the range to the target [34]. The ABL's beacon laser also collected data on atmospheric turbulence. The ABL program will install the kill laser in the aircraft in 2007, and is scheduled to conduct its first airborne missile intercept test in late 2009 [35].

The ABL uses six strategically placed infrared sensors to detect the exhaust trail of a missile. Once a target is detected, the ABL's automated beam control process kicks in. Next, a kilowatt laser tracks the missile and determines a precise aim point. Another kilowatt-class laser then measures atmospheric disturbance, which is then corrected by the adaptive optics system. This information is used to accurately point and focus the high-energy kill laser at the target [36]. Lastly, using a very large telescope located in the nose turret, the beam control system focuses the megawatt kill laser onto a pressurized area of the boosting missile until the directed energy causes the missile structure to fail.

The ABL has at least three known limitations, referred to as the Near Neighbor, Moving Target and Near the Edge failure modes. Near Neighbor failure could occur if multiple missiles are launched from within the ABL's coverage area. This could cause confusion for the ABL in determining which missile to target. Moving Target failure could occur if the ABL has begun engaging the target missile and an object (decoy missile, enemy fighter, etc.) passes between the ABL and the missile target. This could cause the ABL to lose focus on the target missile and track the passing object. Near the Edge failure could occur if the ABL gets a late track on the target missile, which would cause a lag in tracking. The missile would be ahead of or outside of the ABL's tracking field of view.

Despite these failure modes, the ABL still promises to be a defensive force to be reckoned with. Where the ABL falls short, other components of the boost phase BMDS fill the void. These components include the Airborne Tactical Laser, which is a smaller, less power, but more mobile version of the ABL technology.

12 Conclusion

This section highlights the major results of our work in FY07 and discusses followon research to extend these results in FY08.

12.1 High-Assurance Requirements Specification, Validation and Verification Techniques

The research presented in Sections 3 and 4 brings together the Statechart-Assertion and MSC-Assertion formalisms, run-time monitoring, discrete event simulations, and JUnit based test methodology to support a scenario-based, iterative process for specifying, validating, and verifying complex temporal requirements of distributed reactive systems. The Statechart-Assertions and MSC-Assertions are similar to the intuitive and familiar UML MSCs, making them easier to use and understand than text-based temporal assertions of the kind found in literature such as Linear-time Temporal Logic (LTL). Various formal verification techniques suggested in the literature approach the correctness of Web Services temporal behaviors by expressing these temporal properties as LTL statements and subsequent model checking. Most model checkers, like SPIN, do not support specifications with real-life constraints such as real-time and time-series. Moreover, LTL has a rather weak expressive power (LTL is sub-regular). In contrast, MSC Assertions use Java/C++ as an underlying language and therefore enjoy Turing-equivalent descriptive power. The scenario-based, iterative process help ensure the correctness of formal requirements per the modeler's expectations early in the development process, and the use of discrete event simulation in tandem with automatic, JUnit-based, white-box testing and run-time verification to verify the temporal behavior for distributed system prototypes.

One of the challenges in assembling Service Oriented Architecture based BMDS is that the safety of a system can only be evaluated within the context of the system and its operational environment, whereas the various service providers are often developed without consideration of the target system context. Section 2 describes the use of safety-wrappers and safety executives to help safeguard the operation of reusable components that are oblivious of the system context. Safety wrappers monitor the behavior of the re-usable components within the system context, and safety executives help reduce the probability of occurrence of mishaps by providing a centralized point for safety processing: the detection, tolerance and isolation of faults that may result in safety hazards.

We plan to continue our work on MSC-Assertions and run-time execution monitoring to provide new means for engineers to specify time- and safety-critical behaviors in SOA-based SoSes as web services contracts, and develop tools to generate executable code from these contracts for use in service provider selection, integration test automation, and run-time monitors to ensure that the services do behavior correctly and safely in its new environment.

12.2 Run-time Network Security Monitoring

One of the many facets of missile defense network security is the availability of event traces for ongoing network forensic analysis. Instead of logging every events in the missile defense network, we need automation to monitor the network and only log interesting (or suspicious) event sequences. We plan to investigate the use of MSC-Assertions to specify interesting behavior of event sequences and run-time execution monitoring to watch the network traffic and record all event sequences satisfying the specified behaviors.

12.3 Real-Time Java System Evaluation

Section 5 described the results of our experiments with the SUN Java RTS 2.0 beta release that includes the real-time garbage collector. The experiments confirmed the viability of our proposed design pattern for the GIFC that allows developers to use (and re-use) Java code libraries and components that use the heap in the time-constrained applications. We plan to extend our study to the IBM WebSphere real-time Java system and compare the performances between these systems.

12.4 Data Fusion

Data Fusion, the topic of section 6, showed a method of evaluating the performance of multiple radars. By calculating and assigning a score to each sensor track the optimum guide for the interceptor of the threat could be found. The implementation of different types of radar in the vicinity of the missile launch site can be an issue for further investigation. Radars with different parameters, such as transmitted power, frequency, PRF, antenna gain and antenna temperature based on actual sky temperatures for the radar frequency and beam elevation angles of interest will present a more realistic scenario for the data fusion problem. In this situation, the distance between the sensor and the target is not the primary factor for evaluating the data fusion problem. The data fusion problem for the IR sensors based on the likelihood ratio can be studied in a future research. The fusion of the IR sensor data in boost phase is not the primary goal as the function of IR is only to detect the ballistic missile. In the next stages is where the IR sensor can contribute to the tracking of the missile hence the data fusion problem. A study involving deployment of multiple simultaneously launched ballistic missiles is of interest. A network of LEO satellites with IR sensors may be employed to discriminate targets and determine the tracks of multiple targets.

12.5 Ballistic Missile Tracking

The first experiments of Section 7 concluded that the IMM tracking proved superior to Kalman Filter Tracking in the transition from the boost to coast phase of the missile. The later experiments on the Kalman Backfit demonstrated a bias with large uncertainties when determining the missile launch site. The Hyperbolic Backfit gave a reasonably fast and superior result to the Kalman Backfit as well as the Evolutionary Programming and Particle filter backfit. Section 8 discussed appropriateness of the Multiple Hypothesis Tracking to the application ballistic missile tracking. The likeness of the missile data as compared to real-world threat platforms helped to demonstrate the feasibility of the MHT's application. The results show that a technique called linear assignment problem (LAP) used in the implementation of the algorithm is successful in an environment where complex interactions of missile staging, non-linear thrust profiles and sensor noise can significantly degrade the algorithm's performance, especially in multiple target scenarios. Determining the N-best associations and each respective probability was beneficial as it offered a means to expediently identify correct target-to-next-measurement pairings within each scan. The linear assignment approach avoided propagating infeasible hypotheses to later scans and reduced the computational complexity.

Our tracking research this past year has demonstrated that the IMM tracker is capable of producing high-fidelity tracks during boost phase and in the transition from boost to coast phases of a ballistic missile flight trajectory using range and bearing measurements. There are three lines of research arising from our work this year that we wish to pursue.

First, we would like to extend the measurement model to include measurements from alternative sensors, some including Doppler measurements and some with bearing-only measurements, assessing the effect on track error for different sensor suites. A related problem concerning errors generated by preprocessing range-bearing measurements in long range radar data and EKF tracking algorithms bears investigation for the potential effects on tracking errors especially during midcourse.

Second, we would like to use our track estimates to determine the likely target site in order to cue assets that can be used downstream (midcourse attack or terminal defense). Hence, reasonable estimates of both launch point determination and target determination should be generated rapidly for real-time cueing of quick response attacking and defending assets.

Finally, we wish to study the synergism between interceptor guidance and tracking estimation errors to learn not only what guidance algorithms are likely to give the best performance for boost-phase intercept, but also what algorithms and launch strategies can best tolerate tracking and prediction errors, both prior to launch and during midcourse update of the interceptor.

12.6 Sensor Networks

Wireless Sensor Networks (WSN), consisting of nodes with sensing, computational and wireless communications capabilities, can be applied to target detection and tracking, surveillance, monitoring and inventory management. During the past year, we proposed a hybrid, large-scale wireless sensor network (WSN) to support real-time target detection and tracking of multiple ballistic missile threats of all ranges and in all phases of flight.

The data dissemination and real-time tracking require optimization of the network data throughput and end-to-end time-delay. The Area of Interest (AOI) clustering mechanism is introduced, which combines the "content based" feature of the *data centric* routing approach with the principles of in-network data aggregation and clustering. Building upon this AOI mechanism, a data dissemination algorithm suitable for a hybrid large-scale WSN, designed to meet the real-time and accuracy operational requirements imposed by the nature and mission of the WSN without introducing excessive data overhead and increased time-delays is presented. Medium access control (MAC) schemes are investigated since the nature of the proposed WSN necessitates for a shared medium consideration, rather than treating the node interconnections as dedicated point-to-point communication links. A contention based scheme, Carrier Sense Multiple Access (CSMA) MAC, and a contention free approach, Time Division Multiple Access (TDMA) MAC, are examined.

In the previous work, a two-color (MWIR-VLWIR) QWIP FPA sensor was considered. Further investigation into an IR detector with multi-color capabilities of more than two-colors, involving FPAs of larger and different format is proposed. Such a development would cover almost all the IR sensor requirements of a BDMS. Moreover, the examination of modern detector materials such as the Sb-based strained-layer superlattice (type-II SLS) photodetectors is highly recommended since they combine the advantages of both HgCdTe (high quantum efficiency) and QWIP (high uniformity). Additionally, we only addressed the RF sensors briefly. We propose to evaluate specific RF sensors in a hybrid WSN environment in terms of their capabilities for target detection / tracking, kill-assessment as well as background and countermeasure discrimination throughout the trajectory of the ICBM.

The data dissemination mechanism investigated modeled the traffic flow from the sensor nodes to the sink (command and control center), assuming that the AOI clustering mechanism had a limited number of sensor members, a low degree of in-network data aggregation was performed within the AOI, and the target trajectory was in bound toward the sink. Future work may consider modeling multiple traffic flows including weapon assignment flows, expanding the AOI to include more sensors, performing a higher degree of in-network data aggregation involving data fusion techniques and examining the case of multiple sinks.

The performance of the proposed network security mechanism was demonstrated using sequential processing of the associated cryptographic functions and addressed only authentication and confidentiality issues. Future work will include a complete implementation and subsequently fielding of the security mechanism proposed. As part of this follow-on work, the analysis of the TESLA algorithm will be expanded. Additional improvements to the performance of the security mechanism, including parallel processing and data aggregation within the AOI, may be investigated. A complete security solution would also address other security challenges, such as defending against denial of service (DoS) and other types of attacks, detecting misbehaving or rogue nodes, preventing traffic analysis, and optimizing key management techniques.

12.7 Simulation of Electronic Warfare Metrics

In this work we will complete the investigation to characterize the duality or relationship between the network-centric parameters and the warfare performance in the battlespace. For example, the number of nodes (in network space) relates to the situational awareness (in the battlespace) and the number of connections (in network space) relate to the agility (in the battlespace). We will also complete the simulation to quantify the metrics given above and include (for any number of EW nodes), the time dependent network evolution. Questions to be answered include: how can time dependence be included in the simulation to model the flow component which is scaled by the route length? How can time dependence be included in the node capability value? How can coordinated and scripted interactions be quantified and compared using the network metrics.

By using the source rate and observation of the OODA tempo in the analysis of the network robustness insight into the action tempos can be realized. Question to be answered include: What are the necessary action tempos that must be included in the OODA cycle speed analysis? What action tempos correspond to realistic scenario situations such as flight time, speed of force deployment, jammer ON time etc?

When a network centric approach to EW is used, many questions can not be answered in a straightforward formulation. Questions such as how the different degrees of networking impact the strategic, operational and tactical outcomes must be known in order that an optimal network topology can be formed (physical, virtual arrangement of elements). How the network topology will impact the C2 is also of interest since this affects the correct balance of sensors, shooters and network technology. Of utmost importance is the network assurance and how we can quantify the ability of the network to sustain degradation. These issues must be measured in order to improve the understanding of the network-centric approach to EW. A simulation flow chart is shown below in Figure 27.

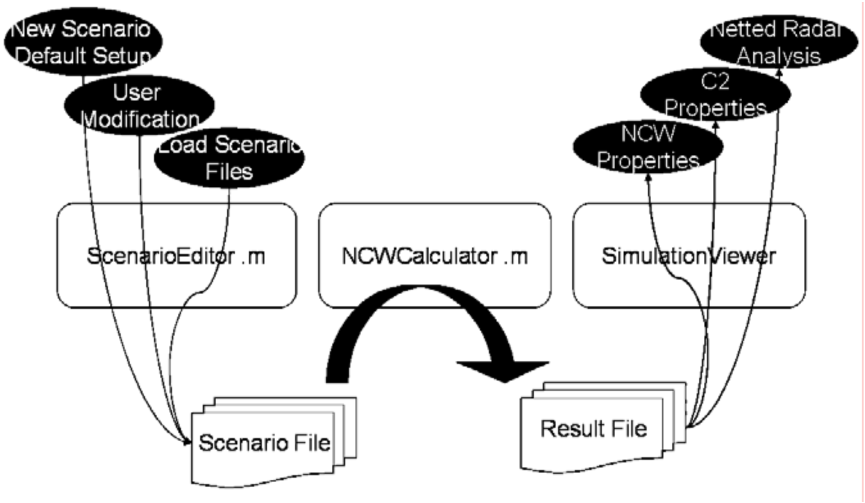


Figure 27: Simulation flowchart [28]

13 References

[1] J. B. Michael and M. Shing, "Safety Assurance of Reconfigurable and Self-Reconfigurable Systems," *Proc. Conf. .Reconfigurable Systems, Microsystems, and Nanotechnology*, Redstone Arsenal, Ala., May 2007.

[2] T. S. Cook, D. Drusinsky and M. Shing, "Specification, Validation and Run-time Monitoring of SOA Based System-of-Systems Temporal Behaviors," *Proc. 2007 IEEE Int'l Conf. System of Systems Engineering*, San Antonio, Tex., April 2007.

[3] D. Drusinsky and M. Shing, "Verifying Distributed Protocols using MSC-Assertions, Run-time Monitoring, and Automatic Test Generation," *Proc. 18th IEEE/IFIP Int'l Workshop Rapid Systems Prototyping*, Porto Alegre, Brazil, June 2007, pp. 82-88.

[4] D. Drusinsky, *Modeling and Verification Using UML Statecharts*, Elsevier, 2006.

[5] M. Auguston, T. S. Cook, D. Drusinsky, J. B. Michael, T. W. Otani, and M. Shing, *Experiments with Sun Java Real-Time System, Part II,* Technical Report, NPS-CS-07-005, Dept. of Computer Science, Naval Postgraduate School, Monterey, Calif., May 2007.

[6] T.S. Cook, D. Drusinsky, J. B. Michael, T. W. Otani, and M. Shing, *Design of Preliminary Experiments with the Sun Java Real-Time System,* Technical Report, NPS-CS-06-010, Dept. of Computer Science, Naval Postgraduate School, Monterey, Calif., May 2006.

[7] D. Patsikas, Track Score Processing of Multiple Dissimilar Sensors, Master's Thesis, Naval Postgraduate School, Monterey, Calif., June 2007.

[8] NASIC, IMPULSE Ballistic Threat Model, Version 1.0, Wright-Patterson Air Force Base, Dayton, Ohio.

[9] R. G. Hutchins and A. San Jose, "IMM Tracking of a Theater Ballistic Missile During Boost Phase," *Signal and Data Processing of Small Targets 1998*, O. Drummond, ed., *Proc. SPIE*, vol. 3373, 1998, pp. 528-537.

[10] A. San Jose, *Theater Ballistic Missile Defense -- Multisensor Fusion, Targeting and Tracking Techniques*, Master's Thesis, Naval Postgraduate School, Monterey, Calif., 1998.

[11] R. G. Hutchins and P.T. Britt, "Trajectory Tracking and Backfitting Techniques Against Theater Ballistic Missiles," *Signal and Data Processing of Small Targets*, O. Drummond, ed., *Proc. SPIE*, vol. 3809, 1999, pp. 532-536.

[12] P. T. Britt, *Kalman Filter Tracking and Backfitting for Boost Phase Theater Ballistic Missiles*, Master's Thesis, Naval Postgraduate School, Monterey, Calif., 1998.

[13] R. G. Hutchins and P.E. Pace, "Improved Trajectory Tracking and Launch Point Determination for Ballistic Missile Defense," *Signal and Data Processing of Small Targets*, O. Drummond, ed., *Proc. SPIE*, vol. 6236, 2006, pp. 62360Y1 - 62360Y8.

[14] H. A. P. Blom, "An Efficient Filter for Abruptly Changing Systems," *Proc. 23rd IEEE Conf. Decision and Control*, Las Vegas, Nev., Dec. 1984, pp. 656-658.

[15] H.A.P. Blom and Y. Bar-Shalom, “The Interacting Multiple Model Algorithm for Systems with Markovian Switching Coefficients,” *IEEE Trans. Automatic Control*, vol. AC-33, no. 8, 1988, pp. 780-783.

[16] M. Yeddanapudi, Y. Bar-Shalom and K. R. Pattipati, “IMM Estimation for Multitarget-Multisensor Air Traffic Surveillance,” *Proc. IEEE*, vol. 85, no. 1, 1997, pp. 80-94.

[17] P. Zarchan, *Tactical and Strategic Missile Guidance*, 4th ed., American Institute of Aeronautics and Astronautics, Reston, Va., 2002.

[18] B. Rakdham, M. Tummala, P.E. Pace, J.B. Michael and Z.P. Pace, “Boost Phase Ballistic Missile Defense Using Multiple Hypothesis Tracking,” *Proc. 2007 IEEE Int’l Conf. System of Systems Engineering*, San Antonio, Tex., Apr. 2007.

[19] R. Danchick and G.E. Newnam, “A Fast Method for Finding the Exact N-best Hypotheses for Multitarget Tracking,” *IEEE Trans. Aerospace and Electronic Systems*, vol. 29, no 2, 1993, pp. 555-560.

[20] B.R. Mahafza, *Radar Systems Analysis and Design Using MATLAB*, Chapman & Hall/CRC, Boca Raton, FL, 2000.

[21] V. Nagarajan, M.R. Chidambara and R.N. Sharma, “New Approach to Improved Detection and Tracking Performance in Track-While-Scan Radars,” *IEE Proc.*, vol. 134, part F, no. 1, Feb. 1987, pp. 93-98.

[22] T.O. Walker III, M. Tummala, J.B. Michael, “A Distributed Medium Access Control Protocol for Wireless Networks of Cooperative Radar Systems,” *Proc. IEEE Int’l Conf. System of Systems Engineering*, San Antonio, Texas, Apr. 2007.

[23] P. Katopodis, G. Katsis, O. Walker, M. Tummala and J.B. Michael, “A Hybrid, Large-scale Wireless Sensor Network for Missile Defense,” *Proc. IEEE Int’l Conf. System of Systems Engineering*, San Antonio, Tex., Apr. 2007.

[24] G. Katsis, Multistage Security Mechanism for Hybrid, Large Scale Wireless Sensor Networks, Master’s Thesis, Naval Postgraduate School, Monterey, Calif., June 2007.

[25] B.Y. Liu, Simulation of Chinese OTH Radar Capabilities Using Internet Space Weather and Radio Propagation Forecasting Software, From Solar Terrestrial Dispatch, Master’s Thesis, Naval Postgraduate School, Monterey, Calif., Sept. 2007.

[26] N.-J. Li, “A Review of Chinese Designed Surveillance Radars -- Past, Present and Future,” *Proc. IEEE Int’l Radar Conf.*, Alexandria, Va., May 1995, pp. 288-293.

[27] Report to Congress Pursuant to the FY2000 National Defense Authorization Act Annual Report on the Military Power of the People’s Republic of China, The - Current and Future Military Strategy of the People’s Republic of China, National Defense Authorization Act for Fiscal Year 2000, Section 1202.

[28] J. Kruse and M. Adkins, “Network Centric Warfare in the U.S. Navy’s 5th Fleet,” *Proc. 38th Hawaii Int’l Conf. System Science*, Big Island, HI Jan. 2005, p. 45a.

[29] F. Stein, J. Garska, and P.L. Mcindoo, “Network Centric Warfare Impact on Army Operations,” *Proc. EUROCOMM 2000 - Information Systems for Enhanced Public Safety and Security*, Munich, Germany, 2000, pp. 288-295.

[30] R. Stapleton-Gray, “A Network That Organizes Itself,” *Defense News,* Oct. 23, 2006.

[31] Y.Q. Chen, Simulation of Network-Enabled Electronic Warfare Metrics to Assess Value of Information and Networking in a General Radar and Communication Topology, Master’s Thesis, Naval Postgraduate School, Monterey, Calif., Sept. 2007.

[32] Missile Defense Agency (MDA), Airborne Laser Fact Sheet, 2006, www.mda.mil.

[33] Janes.com, Boeing/Lockheed Martin/Northrop Grumman YAL-1A Airborne Laser (ABL) Programme (United States), Jane's Information Group, Mar. 2007.

[34] Janes.com, *YAL-1A Airborne Laser*, Jane's Information Group, June 2007.

[35] Boeing Integrated Defense Systems, Boeing-led Airborne Laser Team Actively Tracks Airborne Target, Compensates for Atmospheric Turbulence and Fires Surrogate High-Energy Laser, News Release, July 2007. http://www.boeing.com/news/releases/2007/q3/070716c_nr.html

[36] L. Pullum, M. Darrah, S.T. Skias, K.S. Tso and A.T. Tai, “Developing a data driven prognostic system with limited system information,” *Proc. 8th IEEE Int’l Symp. High Assurance Systems Engineering*, Tampa, FL, Mar. 2004, pp. 281-282.

14 List of Acronyms

1) ABL – Airborne Laser

2) AOI – Area of Interest

3) BMDS – Ballistic Missile Defense System

4) BPEL – Business Processing Execution Language

5) C2BMC – Command and Control, Battle Management, and Communications

6) CR-MAC – Cooperative Radar Medium Access Control

7) CSMA – Carrier Sense Multiple Access

8) DoS – Denial of Service

9) EKF – Extended Kalman Filter

10) EW – Electronic Warfare

11) FMECA – Failure Modes, Effects and Criticality Analysis

12) GEO – Geostationary Earth Orbit

13) GIFC – Global Integrated Fire Control System

14) HF – High-Frequency

15) IMM – Interactive Multiple Model

16) KEI – Kinetic Energy Interceptors

17) KF – Kalman Filter

18) ISR – Intelligence, Surveillance and Reconnaissance

19) LAP – Linear Assignment Problem

20) LEO – Low Earth Orbit

21) LIDAR – Light Detection and Ranging

22) LPRF – Low Pulse Repetition Frequency

23) LTL – Linear-time Temporal Logic

24) MAC – Medium Control Access

25) MDA – Missile Defense Agency

26) MDNT – Missile Defense National Team

27) MEADS – Medium Extended Air Defense System

28) MHT – Multiple Hypothesis Tracking

29) MSC- Message Sequence Chart

30) NCS – Nagarajan, Chidambara, and Sharma

31) OMNeT++ – Objective Modular Network Testbed in C++

32) OODA – Observe, Orient, Decide, Act

33) OTH – Over the Horizon

34) PHA – Preliminary Hazard Analyses

35) PRF – Pulse Repetition Frequency

36) REM – Runtime Execution Monitoring

37) RF – Radio Frequency

38) RTGC – Real-Time Garbage Collector

39) RTS – (Sun Java) Real-Time System

40) SNR – Signal-to-Noise Ratio

41) SOA – Service-Oriented Architecture

42) SoS – System-of-Systems

43) SoSes – Systems-of-Systems

44) SRTM – Safety Requirements Traceability Matrix

45) SSRTM – Software Safety Requirements Traceability Matrix

46) TDMA – Time Division Medium Access

47) THAAD – Terminal High Altitude Area Defense

48) UAV – Unmanned Air Vehicles

49) UML – Unified Modeling Language

50) UML-RT – Unified modeling Language for Real Time

51) WBATG – White-Box Automatic Test Generator

52) WS – Web Services

53) WSN – Wireless Sensor Network

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
8725 John J. Kingman Rd., STE 0944
Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library, Code 52
Naval Postgraduate School
Monterey, CA

3. Research Office, Code 09
Naval Postgraduate School
Monterey, CA

4. Mr. Richard Ritter
Missile Defense Agency
Washington, DC

5. Mr. Michael Young
Missile Defense Agency
Washington, DC

6. Ms. Denise Spencer
Missile Defense Agency
Washington, DC

7. Mr. Steve Hill
Missile Defense Agency
Washington, DC

8. Mr. Jan Young
Missile Defense Agency
Washington, DC

9. Ms. Genell Hausauer
Missile Defense Agency
Washington, DC

10. LTC. Scott LeMay, USAF
Missile Defense Agency
Washington, DC

11. Ms. Deborah Stiltner
Missile Defense National Team
Crystal City, VA

12. Mr. Erik Stein
Missile Defense National Team
Crystal City, VA

13. Mr. Tim Trapp
Missile Defense National Team
Crystal City, VA

14. Dr. Butch Caffall
NASA IV&V Facility
Fairmont, WV

15. Dr. Mikhail Auguston
Naval Postgraduate School
Monterey, CA

16. LTC Thomas Cook
Naval Postgraduate School
Monterey, CA

17. Dr. Doron Drusinsky
Naval Postgraduate School
Monterey, CA

18. Dr. Robert Hutchins
Naval Postgraduate School
Monterey, CA

19. LTJG Panagiotis Katopodis
Naval Postgraduate School
Monterey, CA

20. Dr. Jeffery N. Knorr
Naval Postgraduate School
Monterey, CA

21. Dr. Bret Michael
Naval Postgraduate School
Monterey, CA

22. Dr. Thomas Otani
Naval Postgraduate School
Monterey, CA

23. Dr. Phillip E. Pace
Naval Postgraduate School
Monterey, CA

24. Dr. Man-Tak Shing
Naval Postgraduate School
Monterey, CA

25. Dr. Murali Tummula
Naval Postgraduate School
Monterey, CA

26. CDR Thaddeus Owens Walker
Naval Postgraduate School
Monterey, CA